

**FORSCHUNGSZENTRUM JÜLICH GmbH**  
**Zentralinstitut für Angewandte Mathematik**  
**D-52425 Jülich, Tel. (02461) 61-6402**

Interner Bericht

**Schnelle Wavelet basierte Berechnung von  
langreichweitigen Wechselwirkungen in  
Vielteilchensystemen**

*Silke Wadow*

FZJ-ZAM-IB-2005-04

April 2005

(letzte Änderung: 30.04.2005)



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>9</b>
1.1	Klassische Molekulardynamik . . . . .	9
1.2	Wavelets in MD-Simulationen . . . . .	10
1.3	Ziele . . . . .	12
<b>2</b>	<b>Grundlagen der MD-Simulation</b>	<b>13</b>
2.1	Grundlagen . . . . .	13
2.2	Startkonfiguration . . . . .	13
2.3	Randbedingungen . . . . .	14
2.4	Wechselwirkungen . . . . .	16
2.5	Integratoren . . . . .	18
2.6	Nachbarschaftslisten . . . . .	21
<b>3</b>	<b>Das Verfahren</b>	<b>25</b>
3.1	Nahfeldberechnung mittels PP-Methode . . . . .	26
3.2	Fernfeldberechnung mittels PM-Methode . . . . .	27
3.2.1	Gewichtungsfunktion $\omega_h^i$ . . . . .	29
3.2.2	Einsatz der Wavelets im Verfahren . . . . .	30
<b>4</b>	<b>Wavelets</b>	<b>33</b>
4.1	Einführung . . . . .	33
4.2	Kontinuierliche Wavelet-Transformation . . . . .	34
4.3	Diskrete Wavelet-Transformation . . . . .	36
4.3.1	Multiskalen-Analyse . . . . .	37
4.3.2	Skalierungsfunktion . . . . .	39
4.3.3	Wavelet . . . . .	39
4.4	Schnelle Wavelet-Transformation . . . . .	40
4.4.1	Transformation . . . . .	40
4.4.2	Rekonstruktion . . . . .	42
4.4.3	Komplexität . . . . .	43
4.4.4	Beispiel: Haar-Wavelet . . . . .	44
4.5	Eigenschaften von Wavelets . . . . .	45
4.6	Daubechies Wavelet Filterkoeffizienten . . . . .	47
4.7	Wavelet-Thresholding . . . . .	50

<b>5</b>	<b>Ergebnisse</b>	<b>53</b>
5.1	Abschätzung des Diskretisierungsfehlers . . . . .	53
5.2	Kompressionsverhalten . . . . .	58
5.3	Gesamtfehler . . . . .	60
5.4	Laufzeitverhalten . . . . .	62
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>67</b>
<b>A</b>	<b>Verwendete Wavelets</b>	<b>71</b>
<b>B</b>	<b>Abstandsmatrix</b>	<b>75</b>
<b>C</b>	<b>Quellcode - FWT</b>	<b>81</b>
<b>D</b>	<b>Notationen</b>	<b>83</b>

# Abbildungsverzeichnis

1.1	Propagation in einer MD-Simulation . . . . .	9
2.1	Randbedingungen . . . . .	15
2.2	Periodische Randbedingungen . . . . .	16
2.3	Verlet - Liste . . . . .	22
3.1	Nahfeldberechnung - PP-Methode . . . . .	26
3.2	Interpolation auf Gitterpunkte . . . . .	28
3.3	NGP-Schema - Instabilität . . . . .	29
3.4	CIC-Schema - Zuordnung der Ladungsanteile . . . . .	30
4.1	Translation und Dilatation des Mexikanerhuts . . . . .	35
4.2	Translation und Dilatation des <i>Daub<sub>4</sub></i> -Wavelets . . . . .	35
4.3	Multiskalenanalyse . . . . .	37
4.4	Schnelle Wavelet-Analyse . . . . .	42
4.5	Pyramiden-Algorithmus . . . . .	42
4.6	Schnelle Wavelet-Rekonstruktion . . . . .	43
4.7	Filterdarstellung . . . . .	48
4.8	Hard- und Soft-Thresholding . . . . .	50
5.1	Initialisierungsaufwand . . . . .	54
5.2	Aufwand pro Zeitschritt . . . . .	54
5.3	Fehlerverhalten der Interpolation - period. Randbedingungen . . . . .	55
5.4	Diskretisierungsfehler - period. Randbedingungen . . . . .	56
5.5	Fehlerverhalten der Interpolation - offenes System . . . . .	57
5.6	Diskretisierungsfehler - offenes System . . . . .	57
5.7	Vergleich mit explizitem Verfahren . . . . .	58
5.8	Kompressionsraten verschiedener Levels . . . . .	60
5.9	Kompressionsraten verschiedener Wavelettypen . . . . .	61
5.10	Threshold-Verfahren - Fehlerverhalten . . . . .	61
5.11	Gesamtfehler . . . . .	62
5.12	Interpolation - Laufzeit . . . . .	63
5.13	Nahfeldberechnung - Laufzeit . . . . .	64
5.14	Matrix-Vector Multiplikation - Laufzeit . . . . .	64
5.15	Zeitschritt - Laufzeit . . . . .	65

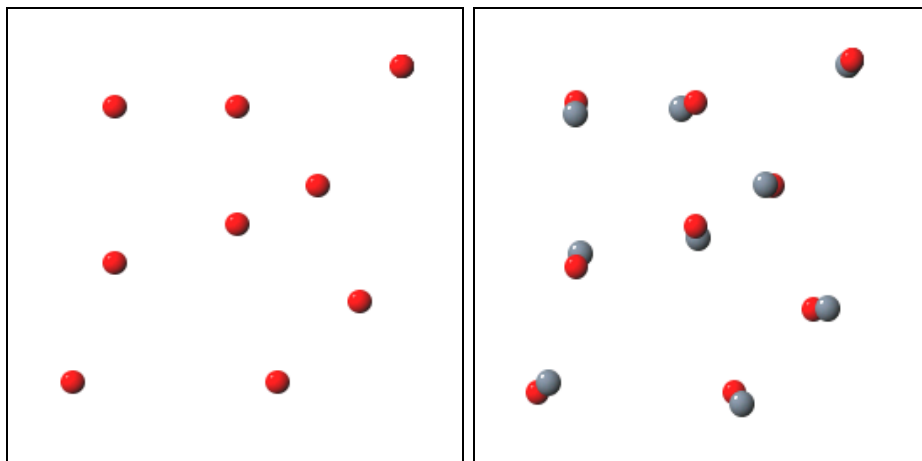
A.1	<i>Daub2</i> -Wavelet . . . . .	71
A.2	<i>Daub4</i> -Wavelet . . . . .	71
A.3	<i>Daub12</i> -Wavelet . . . . .	72
A.4	<i>Daub20</i> -Wavelet . . . . .	72
B.1	Abstandsmatrix . . . . .	75
B.2	Wavelettransformierte Abstandsmatrix . . . . .	75
B.3	Wavelettransformierte mit Hard-Thresholding . . . . .	76
B.4	Rücktransformierte Abstandsmatrix 1 . . . . .	76
B.5	Wavelettransformierte mit Soft-Thresholding . . . . .	77
B.6	Rücktransformierte Abstandsmatrix 2 . . . . .	77
B.7	Fehlerverteilung Hard-Thresholding . . . . .	78
B.8	Fehlerverteilung Soft-Thresholding . . . . .	78
B.9	Streifenstruktur der Matrix . . . . .	79

# Kapitel 1

## Einleitung

### 1.1 Klassische Molekulardynamik

Die klassische Molekulardynamik (MD) beschäftigt sich mit den Bewegungen von miteinander wechselwirkenden Teilchen eines Systems. In jedem Zeitschritt werden über die Wechselwirkungspotentiale die Kräfte auf alle Teilchen berechnet, woraus sich dann die Geschwindigkeitsänderung und aus der Geschwindigkeit schließlich eine Positionsänderung für jedes Teilchen ergibt. Dies wird in jedem Zeitschritt für alle Atome/Teilchen durchgeführt. Eine schematische Darstellung ist in Abbildung 1.1 zu sehen.



**Abbildung 1.1:** Schematische Darstellung einer MD-Simulation: Im linken Bild sind die Teilchen im  $n$ -ten Zeitschritt zu sehen, im rechten Bild stellen die roten Kugeln die Positionen der Teilchen im  $n$ -ten und die grauen Kugeln die Positionen im darauf folgenden Zeitschritt dar.

Obwohl das Prinzip der klassischen Molekulardynamik sehr einfach ist, wird diese Methode erst seit etwa 40 Jahren zur numerischen Simulation verwendet. Die Ursache hierfür ist der enorme Bedarf an Rechenleistung, der selbst zur Berech-

nung von Systemen mit nur wenigen Atomen nötig ist. Erst mit der Einführung der elektronischen Rechner wurde daher die praktische Anwendung möglich. Bis zu jenem Zeitpunkt war die oben beschriebene schrittweise Lösung der Newtonschen Bewegungsgleichungen für eine große Anzahl von Atomen praktisch nicht durchführbar.

Um die Anzahl der zu berechnenden Kräfte nicht zu groß werden zu lassen, muss man in der Regel die Reichweite, bis zu der Wechselwirkungen eingehen, begrenzen, was wiederum dazu führt, dass weit reichende Kräfte wie z.B. die Coulomb-Wechselwirkungen gesondert behandelt werden müssen. Erst das “Abschneiden” der Potentiale beim Cutoff-Radius  $r_c$  gestattet es, eine große Anzahl von Atomen molekulardynamisch zu behandeln (vgl. Kapitel 2.6).

In diesem Fall kann die Simulation auch parallelisiert werden: Jedes Atom kann nur mit den Atomen in einer Cutoff-Kugel mit Radius  $r_c$  wechselwirken. Zerlegt man das System in Zellen, so kann ein Atom nur mit anderen Atomen aus derselben Zelle oder aus den direkten Nachbarzellen wechselwirken. Diese Zellen kann man geschickt auf mehrere Prozessoren verteilen und so die gesamte Simulationszelle in kleinere Zellen untergliedern, die von je einem Prozessor bearbeitet werden.

Die numerische Integration der Bewegungsgleichungen kann mit vielen allgemein bekannten Integratoren wie z.B. den Algorithmen von Gauß, Verlet oder Runge-Kutta durchgeführt werden (vgl. Kapitel 2.5). Weitere Variationsmöglichkeiten bieten die Wahl der Randbedingungen. Periodische Randbedingungen können das betrachtete System in eine oder mehrere Richtungen unendlich lang machen, mit anderen Randbedingungen dagegen kann man Kräfte und Momente auf das System aufbringen (vgl. Kapitel 2.3).

Der Erfolg von klassischen molekulardynamischen Simulationen hängt entscheidend von der Qualität der verwendeten effektiven Potentiale ab. Während für einige Systeme Potentiale bekannt sind, die das Verhalten äußerst gut beschreiben, fehlen diese Informationen für andere Strukturen. In diesem Fall ist man auf die Verwendung von Modellpotentialen angewiesen.

Das Problem mit den effektiven Potentialen wird elegant umgangen, indem die Kräfte aus ersten Prinzipien, also den quantenmechanischen Wechselwirkungen aller beteiligten Teilchen, berechnet werden.

## 1.2 Wavelets in MD-Simulationen

Wie bereits erwähnt, müssen in jedem Iterationsschritt einer MD-Simulation die Kräfte auf jedes Teilchen bestimmt werden. In der hier vorliegenden Arbeit werden ausschließlich langreichweitige elektrostatische Wechselwirkungen betrach-



tet. Die Berechnungen der Kräfte können kompakt durch eine Matrix-Vektor-Multiplikation dargestellt werden, wobei in dem Vektor die Ladungen der Teilchen eingetragen sind und in der quadratischen Matrix die inversen Abstände der Teilchen zueinander.

Rein technisch kann bei der Durchführung der Matrix-Vektor-Multiplikation auf jene Koeffizienten verzichtet werden, welche Null oder vernachlässigbar klein sind. Man muss diese weder abspeichern noch irgendwelche arithmetischen Operationen mit ihnen durchführen. Besteht der Großteil der Matrix aus Nullen - man spricht von einer dünn besetzten Matrix - so reduziert sich der Gesamtaufwand der Matrix-Vektor-Multiplikation erheblich.

Mit Hilfe einer geeigneten Wavelet-Transformation kann man diese voll besetzte Abstandsmatrix in eine dünn besetzte transformieren. Die Wavelet-Transformation wird ausführlich in Kapitel 4 diskutiert. Um aus der MD-Simulation allerdings verwertbare Ergebnisse zu gewinnen, muss der Ladungsvektor ebenfalls Wavelet-transformiert und das Ergebnis der Matrix-Vektor-Multiplikation, das Potential, wieder zurücktransformiert werden.

Vorteile bringt dieses Verfahren daher nur dann, wenn sich die Matrix nicht in jedem Zeitschritt ändert, da in diesem Fall die Rechenzeiterparnis bei der Matrix-Vektor-Multiplikation erheblich größer ist als der Mehraufwand, der bei der Durchführung der Wavelet-Transformation entsteht. Da die Teilchen in einer MD-Simulation allerdings durch den Raum propagieren, ändern sich in jedem Zeitschritt die Abstände der Teilchen zueinander, so dass eine Wavelet-Transformation auf den ersten Blick einen unnötigen Mehraufwand bedeutet.

Wird aber über das Simulationsgebiet ein festes Gitter gelegt, die Kräfte zunächst an den Gitterpunkten bestimmt und schließlich von den Gitterpunkten auf die Orte der Teilchen interpoliert, so bringt eine geschickte Wavelet-Transformation eine erhebliche Performance-Verbesserung mit sich, da sich die Abstände der Gitterpunkte zueinander über die gesamte Simulationsdauer nicht ändern. Eine derartige Methode wird auch als Particle-Mesh Methode bezeichnet.

Um die Fehler möglichst klein zu halten, werden lediglich die Kräfte aus dem Fernfeld auf diese Weise ermittelt, d.h. nur die Kräfte, die von genügend weit entfernten Teilchen stammen. Das Nahfeld, also die Kräfte, die von Wechselwirkungen von Teilchen aus der unmittelbaren Umgebung herrühren, wird mit einem explizitem Verfahren berechnet. Wie fein das Gitter gewählt werden muss, damit der Fehler unterhalb einer bestimmten Fehlerschranke bleibt, wird in Kapitel 4 aufgezeigt.

## 1.3 Ziele

Inhalt und Ziel der vorliegenden Arbeit ist die Entwicklung und Untersuchung eines Verfahrens, das den Aufwand zur Behandlung von Vielteilchensystemen mit langreichweitigen elektrostatischen Wechselwirkungen so weit reduziert, dass selbst Systeme mit Hunderttausenden von Teilchen behandelt werden können. Das Verfahren wird den ursprünglichen Aufwand von  $O(N^2)$  auf eine lineare und somit optimale Komplexitätsordnung von  $O(N)$  verringern.

Eine Reduzierung sowohl der Komplexitätsordnung als auch des Speicherplatzbedarfs ist an dieser Stelle aus mehreren Gründen sehr sinnvoll. Zum einen sind allein mehrere Gigabyte an Speicherkapazität nötig, um alle benötigten Informationen im Hauptspeicher zu halten, was besonders wichtig ist, um ein mehrfaches, zeitintensives Ein- und Auslesen von der Festplatte oder anderen externen Speichermedien zu vermeiden. Zum anderen weist das Problem eine sehr hohe Komplexität von  $O(N^2)$  auf.

Damit sichere numerische Ergebnisse erzielt werden, muss bei der Verringerung des rechnerischen Aufwands darauf geachtet werden, dass eine ausreichende Genauigkeit erreicht wird, d.h. der Gesamtfehler muss hinreichend klein sein. Der Gesamtfehler setzt sich aus dem Verfahrensfehler (Diskretisierungsfehler), dem Fehler aus der Wavelet-Transformation - im Folgenden Kompressionsfehler genannt - und dem maschinellen Rundungsfehler zusammen.

# Kapitel 2

## Grundlagen der MD-Simulation

### 2.1 Grundlagen

Die Molekulardynamik (MD) ist eine Simulationsmethode, bei der die Zeit-Ort-Entwicklung von miteinander wechselwirkenden Atomen durch Integration ihrer Bewegungsgleichungen berechnet wird. Eine MD-Simulation besteht im Wesentlichen aus der Initialisierung des Systems, der Equilibrierung, der Evolution durch Integration der Bewegungsgleichungen und der Berechnung der interessierenden Daten. Der anschaulichste und einfachste Fall ist die Dynamik harter Kugeln, da hier nur vollkommen elastische Stöße auftreten.

### 2.2 Startkonfiguration

In der Startkonfiguration werden zum einen sämtliche für die Simulation relevanten Parameter gesetzt wie z.B. die Teilchenanzahl  $N$  oder die Packungsdichte  $\rho$ , zum anderen werden die Startposition und die Charakteristika von jedem einzelnen Teilchen festgelegt.

Aus der Packungsdichte kann ein weiterer wichtiger Parameter berechnet werden: die Länge  $L$  der Simulationsbox.

$$L = \left( \frac{N}{\rho} \right)^{1/3}$$

Es gibt verschiedene Möglichkeiten, die Startpositionen zu setzen. Die einfachste Methode ist, die Startkonfiguration von einer vorherigen Simulation zu übernehmen. Allerdings muss zumindest für die erste Simulation eine Startkonfiguration entweder per Hand eingegeben oder nach einem Schema generiert werden. Daher ist es ratsam, sich Gedanken darüber zu machen, wie man Startpositionen und -geschwindigkeiten bestimmen kann.

Die Startpositionen können z.B. zufällig verteilt sein, so dass man sie mit Hilfe eines Zufallszahlengenerators setzen kann. Diese Vorgehensweise bringt aber Probleme mit sich, denn es ist keineswegs gewährleistet, dass sich keine Teilchen in der Simulation überlappen. Besonders bei Simulationen harter Kugeln hat dieses Verfahren katastrophale Folgen, da sich harte Kugeln per Definition nicht penetrieren können. Aber auch bei Simulationen mit inelastischen Stößen treten Probleme auf, wenn sich die Teilchen bereits in der Startkonfiguration gegenseitig durchdringen.

Um solche Effekte zu vermeiden, werden die Teilchen nach einem regelmäßigen Muster wie in einer Kristallstruktur angeordnet. Diese Kristallstruktur kann durch periodisches Wiederholen einer Einheitszelle aufgebaut werden. Allerdings ist diese Anordnung der Teilchen eher atypisch, so dass vor der eigentlichen Simulation zunächst einige Zeitschritte durchgeführt werden sollten, um von der atypischen, geordneten Verteilung zu einer natürlichen, ungeordneten zu gelangen (Equilibrierung).

Neben den Startpositionen sind noch weitere Charakteristika der Teilchen festzulegen wie z.B. Masse, Durchmesser, Ladung und Geschwindigkeit. Welche Charakteristika besetzt werden müssen, hängt von der Art der Simulation ab. Im Fall der harten Kugeln sind lediglich Masse, Durchmesser und Geschwindigkeit von Interesse. Die Parameter Masse und Durchmesser können durchaus für alle Teilchen einheitlich gewählt werden. Die Startgeschwindigkeit sollte aber nicht für jedes Teilchen gleich sein. Man kann sie entweder aus einer früheren Simulation übernehmen oder mit Hilfe eines Zufallszahlengenerators bestimmen.

Bei der Bestimmung mit Hilfe eines Zufallszahlengenerators sollte man beachten: Die Geschwindigkeit der Teilchen ist nicht gleichverteilt, sondern in jeder Koordinate normalverteilt. Außerdem sollte der Gesamtimpuls  $\vec{P}$  des Systems Null sein, d.h.

$$\vec{P} = \sum_{i=1}^N m_i \vec{v}_i = 0,$$

wobei  $m_i$  die Masse und  $\vec{v}_i$  die Geschwindigkeit von Teilchen  $i$  ist [5].

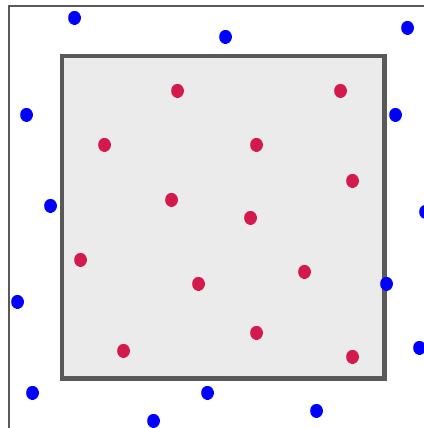
## 2.3 Randbedingungen

Randbedingungen legen fest, wie an den Rändern der Simulationsbox verfahren wird. Bestehen die Ränder des Simulationsgebietes aus Wänden, so können die Teilchen diese nicht durchdringen und prallen von diesen ab. Sie bleiben somit in der Simulationsbox.

Da es oftmals nur von Interesse ist, was im Zentrum passiert, sind Randeffekte möglichst auszuschließen. Um dies zu erreichen, könnten einfach keine Wände in

die Simulation eingebaut werden, d.h. ein Teilchen kann unendlich weit in jede Richtung propagieren. Hier ist allerdings das Problem, dass mit der Zeit immer mehr Teilchen aus der Simulationsbox heraus wandern, so dass am Ende keine Teilchen mehr in der Box übrig bleiben, über die eine Aussage gemacht werden kann. Dieses Hinauswandern der Teilchen entspricht einem nicht geschlossenen System. Der Energie- sowie der Impulserhaltungssatz hat in einem solchen Fall innerhalb des Simulationsgebietes keine Gültigkeit.

Um die Anzahl der Teilchen in der Simulation konstant zu halten, sind Wände daher wichtig. Nun kann man natürlich einfach ein größeres System simulieren und nur in dem Bereich, wo keine Randeffekte auftreten, die interessierenden Größen bestimmen (dargestellt in Abbildung 2.1).

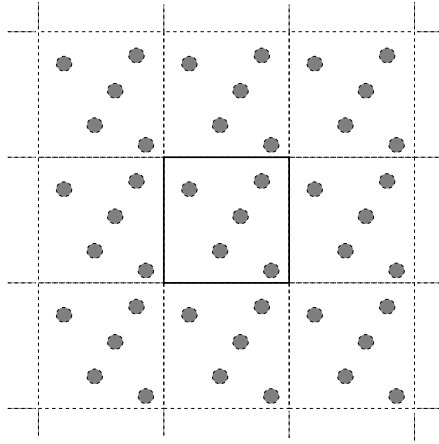


**Abbildung 2.1:** Für alle Teilchen werden die Wechselwirkungen berechnet, jedoch nur die Teilchen aus dem grau unterlegten Bereich werden für die Berechnung der interessierenden Größen herangezogen.

Da sich Randeffekte über Distanzen von mehreren Atomdurchmessern Längen erstrecken und ein  $N$ -Körperproblem eine Komplexität von  $O(N^2)$  aufweist, ist es allerdings sehr zeitaufwendig, für Teilchen die Wechselwirkungen zu berechnen, über die keine Aussage gemacht werden kann. Abgesehen von dem unnötigen Mehraufwand hat man immer noch das Problem, dass die Anzahl der Teilchen, über die eine Aussage getroffen werden kann, nicht in jedem Zeitschritt konstant bleibt.

Damit man nur die interessierenden Teilchen ohne Randeffekte simulieren muss, wurden periodische Randbedingungen eingeführt. Dies bedeutet, dass sich das System in alle Richtungen periodisch fortsetzt. Wandert ein Teilchen links aus der Simulationsbox, so wandert zeitgleich ein Teilchen mit denselben Charakteristika von rechts in die Simulationszelle hinein. Wichtig hierbei ist, dass die Länge der Simulationsbox nicht zu klein gewählt wird. Sie muss größer sein als der Cutoff Radius [5].

Normalerweise werden um die Simulationsbox - bei periodischen Randbedingungen Zentralbox genannt - 26 Bildzellen gelegt, die eine Kopie der Zentralbox sind. Die Position der Bildteilchen  $\vec{r}_{\vec{\alpha}}$  erhält man durch  $\vec{r}_{\vec{\alpha}} = \vec{r} + \vec{\alpha}L$ , wobei  $L$  die Länge der Zentralbox,  $\vec{r}$  die Position der Teilchen innerhalb der Zentralbox und  $\vec{\alpha}$  der Vektor ist, der vom Ursprung der Zentralbox zum Ursprung der jeweiligen Bildbox zeigt und dessen Komponenten entweder -1, 0 oder 1 sind. Eine schematische Darstellung ist in Abbildung 2.2 dargestellt.



**Abbildung 2.2:** Schematische Darstellung periodischer Randbedingungen in 2D: Um die Zentralbox liegen acht identische Bildboxen.

Durch die Bildboxen wird das ursprüngliche System von  $N$  Teilchen künstlich aufgebläht, auch wenn hierdurch kein zusätzlicher Speicherplatz benötigt wird. Für ein Teilchen der Zentralbox bedeutet dies, dass es mit den  $N - 1$  nächst gelegenen Teilchen interagiert. Dieses Prinzip wird als minimum image convention bezeichnet. Probleme bereitet diese Vorgehensweise allerdings bei harten Kugeln, da der Simulation harter Kugeln ein entartetes Potential zu Grunde liegt.

## 2.4 Wechselwirkungen

Die Bewegungen der Teilchen im System können mit den drei Newtonschen Axiomen ausgedrückt werden, wobei die zentrale Bewegungsgleichung der Newtonschen Mechanik das zweite Newtonsche Axiom  $\vec{F}(t) = m \cdot \ddot{\vec{r}}(t)$  ist, eine Differentialgleichung zweiter Ordnung, die den Zusammenhang zwischen der Kraft  $\vec{F}$  auf einen Massenpunkt  $m$  und der Beschleunigung  $\ddot{\vec{r}}$  des Massenpunktes beschreibt.

Alternativ kann diese Gleichung auch in der Hamiltonschen Mechanik formuliert werden. Die Bewegungsgleichungen sind in diesem Fall im Phasenraum gegeben, der durch die unabhängigen Koordinaten  $\vec{q}_j$  und zusätzlich durch die

von den Koordinaten unabhängigen kanonischen Impulsen  $\vec{p}_j$  aufgespannt wird.

Die Hamilton-Funktion  $H$  ist definiert durch

$$H(\vec{q}, \vec{p}, t) = \sum_{i=1}^N \dot{\vec{q}}_i \vec{p}_i - L(\vec{q}, \dot{\vec{q}}, t) \quad ,$$

wobei  $L(\vec{q}, \dot{\vec{q}}, t)$  die Lagrange-Funktion ist [3]. In holonom-skleronomen Systemen, wie sie hier vorliegen, beschreibt die Hamilton-Funktion die Gesamtenergie  $E$  des Systems, welche sich als Summe der kinetischen Energie  $K$  und der potentiellen Energie  $U$  darstellen lässt.

$$\begin{aligned} H &= K + U \\ &= \frac{1}{2} \sum_{i=1}^N \frac{\vec{p}_i^2}{m_i} + U \\ &= E \end{aligned}$$

Das totale Differential  $dH$  liefert die Bewegungsgleichungen in der Formulierung von Hamilton (kanonische Gleichungen) [3, 5].

$$\begin{aligned} \dot{\vec{q}}_i &= \frac{\partial H}{\partial \vec{p}_i} \rightarrow \dot{\vec{r}}_i = \frac{\partial H}{\partial \vec{p}_i} = \frac{\vec{p}_i}{m_i} \\ \dot{\vec{p}}_i &= \frac{\partial H}{\partial \vec{q}_i} \rightarrow \dot{\vec{p}}_i = \frac{\partial H}{\partial \vec{r}_i} = -\frac{\partial U}{\partial \vec{r}_i} = \vec{F}_i \end{aligned}$$

Die Kraft, die auf ein Teilchen einwirkt, wird durch den negativen Gradienten der potentiellen Energie  $U$  bestimmt. Die potentielle Energie kann in mehrere Anteile aufgespalten werden, welche von unterschiedlichen Wechselwirkungsarten herrühren. Diese Anteile können von externen Feldern (1) stammen oder von einer Wechselwirkung zwischen zwei (2), drei (3) oder mehreren Teilchen kommen.

$$U(\vec{r}^N) = \underbrace{\sum_i u^{(1)}(\vec{r}_i)}_1 + \underbrace{\sum_i \sum_{j < i} u^{(2)}(\vec{r}_i, \vec{r}_j)}_2 + \underbrace{\sum_i \sum_{j < i} \sum_{k < j} u^{(3)}(\vec{r}_i, \vec{r}_j, \vec{r}_k)}_3 + \dots$$

In der Regel werden alle Energiebeiträge, die von Wechselwirkungen zwischen mehr als zwei Teilchen herrühren, vernachlässigt. Der Energiebeitrag aus einer Wechselwirkung zwischen zwei Atomen kann noch weiter in die gebundenen intramolekularen und die ungebundenen intermolekularen Wechselwirkungen zerlegt werden.

Innerhalb eines Moleküls kann sich der Abstand zwischen zwei Atomen verringern oder aber auch vergrößern (stretch). Betrachtet man z.B. ein Sauerstoffmolekül, so variiert der Abstand zwischen den beiden Sauerstoffatomen ständig

in schneller Abfolge. Die Lage von den Atomen innerhalb eines Moleküls kann sich ebenfalls verändern durch die Drehung eines Atoms (torsion) oder durch eine Winkelveränderung von Bindungen zueinander (bend). Ein Beispiel für den letzteren Fall ist ein Wassermolekül, bei dem sich der Abstand der Wasserstoffatome zueinander verändert, nicht aber die Abstände zum Sauerstoffatom. Ein einfaches Beispiel für Torsion liegt beim Wasserstoffperoxid-Molekül vor ( $\text{H}_2\text{O}_2$ ). Dreht sich ein Sauerstoffatom längs der Verbindung zum anderen Sauerstoffatom, so ändert das zugehörige Wasserstoffatom seine Position und somit seine Lage zum anderen Wasserstoffatom [5].

An intermolekularen Wechselwirkungen zwischen zwei Molekülen sind vor allem die van der Waals Kräfte, die Polarisierung und die elektrostatischen Kräfte zu nennen.

Berücksichtigt man all diese Energiebeiträge, so wird der Rechenaufwand für eine Simulation erheblich vergrößert. Daher werden die schnellen intramolekularen Wechselwirkungen nicht betrachtet. Besonders auf die Einbeziehung von Wasserstoffatomen wird verzichtet. Aufgrund dieser Vernachlässigungen kann man Moleküle wie ein einzelnes großes Teilchen behandeln, wodurch sich die Teilchenanzahl in der Simulation erheblich reduziert. Ebenfalls wird die Polarisierung nicht mit in die Berechnungen einbezogen, da sie zu rechenzeitintensiven Wechselwirkungen zwischen mehr als zwei Teilchen führt.

Neben der Unterscheidung in intra- und intermolekularen Wechselwirkungen gibt es noch die Unterteilung in lang- und kurzreichweitige Wechselwirkungen. Kurzreichweitige Wechselwirkungen bestehen zwischen einem induzierten Dipol und einer Ladung, einem Dipol oder einem induzierten Dipol. Das Potential, welches durch die Wechselwirkungen entsteht, fällt hier mindestens mit  $1/r_{ij}^{d+1}$  ab, wobei  $r_{ij}$  der Abstand zwischen den Teilchen  $i$  und  $j$  und  $d$  die Dimension ist. Alle Wechselwirkungen, deren Potentiale langsamer abnehmen, sind langreichweitig.

In der vorliegenden Arbeit werden langreichweitige Coulomb Wechselwirkungen betrachtet. Das Potential nimmt mit  $1/r_{ij}$  ab. Somit wird das Potential  $\Phi_i$  und die Kraft  $\vec{F}_i$  auf ein Teilchen  $i$  berechnet durch

$$\begin{aligned}\Phi_i &= \sum_j \frac{q_j}{|\vec{r}_i - \vec{r}_j|} \quad \text{und} \\ \vec{F}_i &= q_i \sum_j \frac{q_j}{|\vec{r}_i - \vec{r}_j|^2} (\vec{r}_i - \vec{r}_j).\end{aligned}$$

## 2.5 Integratoren

Der zeitaufwendigste Teil einer MD-Simulation besteht in der Integration der Bewegungsgleichungen. Daher sollte ein Integrator sehr schnell sein. Neben der



Schnelligkeit spielen hohe Genauigkeit, hohe Stabilität auch für große Zeitschritte, Zeitreversibilität und die Fähigkeit, keine Änderung der Gesamtenergie und des Gesamtimpulses des Systems zuzulassen, eine wichtige Rolle [5].

### Euler Algorithmus

Mit Hilfe der Reihenentwicklung nach Taylor erhält man einen ersten Ansatz:

$$\begin{aligned}\vec{r}(t + \Delta t) &= \sum_{i=0}^{\infty} \frac{1}{i!} \vec{r}^{(i)}(t) \cdot \Delta t^i \\ &= \vec{r}(t) + \vec{v}(t) \cdot \Delta t + \frac{\vec{F}(t)}{2m} \cdot \Delta t^2 + O(\Delta t^3)\end{aligned}\quad (2.1)$$

$$\vec{v}(t + \Delta t) = \vec{v}(t) + \frac{\vec{F}(t)}{m} \cdot \Delta t + O(\Delta t^2) \quad (2.2)$$

Dieser Ansatz führt zum Euler Algorithmus. Die Position im neuen Zeitschritt wird durch die Position, Geschwindigkeit und Beschleunigung zum Zeitpunkt  $t$  ermittelt. Dieser Integrator weist allerdings eine starke Energiedrift auf und ist auch nicht zeitreversibel.

### Verlet Algorithmus

Führt man die Taylorentwicklung sowohl für einen Zeitpunkt in der Zukunft (2.3) als auch in der Vergangenheit (2.4) durch und addiert beide entstehenden Gleichungen, so erhält man Gleichung (2.5).

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t) \cdot \Delta t + \frac{\vec{F}(t)}{2m} \cdot \Delta t^2 + \dots \quad (2.3)$$

$$\vec{r}(t - \Delta t) = \vec{r}(t) - \vec{v}(t) \cdot \Delta t + \frac{\vec{F}(t)}{2m} \cdot \Delta t^2 - \dots \quad (2.4)$$

$$\vec{r}(t + \Delta t) = 2\vec{r}(t) - \vec{r}(t - \Delta t) + \frac{\vec{F}(t)}{m} \cdot \Delta t^2 + O(\Delta t^4) \quad (2.5)$$

Die Geschwindigkeit zum Zeitpunkt  $t$  lässt sich über ein Differenzenschema zweiter Ordnung bestimmen.

$$\vec{v}(t) = \frac{\vec{r}(t + \Delta t) - \vec{r}(t - \Delta t)}{2\Delta t} + O(\Delta t^2)$$

Es handelt sich hierbei um den Verlet Algorithmus. Aus den Positionen der Teilchen zum aktuellen Zeitschritt werden die aktuellen Kräfte auf die Teilchen berechnet. Zusammen mit den Kräften und den Positionen der Teilchen sowohl zum aktuellen als auch zum vorhergehenden Zeitschritt werden die Positionen im nächsten Zeitschritt ermittelt. Da in der Startphase nur die Position der Teilchen zum Startzeitpunkt zur Verfügung steht, muss zunächst ein Zeitschritt mit einem

anderen Verfahren durchgeführt werden, um so die Positionen zu zwei verschiedenen Zeitpunkten zu erhalten.

Dieser Integrator ist zwar zeitreversibel und weist auch keine Energiedrift auf, benötigt aber die Position der Teilchen zu zwei verschiedenen Zeitpunkten und bestimmt die Geschwindigkeit zeitversetzt. Nach einer Startphase kann man allerdings die Geschwindigkeiten zum gleichen Zeitpunkt durch Extrapolation ermitteln.

### Leapfrog Algorithmus

Das Problem des Verlet Algorithmus, Kenntnis von den Positionen an zwei aufeinanderfolgenden Zeitpunkten zu benötigen, löst der Leapfrog Algorithmus, indem die Geschwindigkeiten zu “Halbschritten” definiert werden:

$$\begin{aligned}\vec{v}(t - \Delta t/2) &= \frac{\vec{r}(t) - \vec{r}(t - \Delta t)}{\Delta t} \\ \vec{v}(t + \Delta t/2) &= \frac{\vec{r}(t + \Delta t) - \vec{r}(t)}{\Delta t}\end{aligned}$$

Aus der letzten Gleichung folgt die Integrationsvorschrift:

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \Delta t \cdot \vec{v}(t + \Delta t/2)$$

Formt man den Verlet Algorithmus (Gleichung (2.5)) um, so erhält man:

$$\begin{aligned}\vec{r}(t + \Delta t) - \vec{r}(t) &= \vec{r}(t) - \vec{r}(t - \Delta t) + \frac{\vec{F}(t)}{m} \cdot \Delta t^2 + O(\Delta t^4) \\ \Rightarrow \vec{v}(t + \Delta t/2) \cdot \Delta t &= \vec{v}(t - \Delta t/2) \cdot \Delta t + \frac{\vec{F}(t)}{m} \cdot \Delta t^2 + O(\Delta t^4)\end{aligned}$$

Dies bedeutet, dass man beim Leapfrog Algorithmus für jedes Teilchen zuerst aus der aktuellen Position die aktuelle Kraft berechnet und danach aus der aktuellen Kraft und der Geschwindigkeit in dem vorherigen “Halbschritt” die Position im nächsten Zeitschritt bestimmt.

Der Vorteil gegenüber dem Verlet Algorithmus ist, dass die Position nur noch zu einem Zeitpunkt bekannt sein muss. Nachteilig ist allerdings, dass der Wert für die Geschwindigkeit immer noch nicht zum selben Zeitpunkt verfügbar ist. Zum Starten benötigt man also auch weiterhin ein anderes Verfahren mit halber Schrittweite, welches die Geschwindigkeit zu den “Halbschritten” berechnet. Die Geschwindigkeit zum Zeitpunkt  $t$  kann bei diesem Integrator durch Interpolation ermittelt werden.

### Velocity Verlet Algorithmus

Die Berechnung der Positionen zum Zeitpunkt  $t + \Delta t$  ist beim Leapfrog Algorithmus bereits sehr effizient, doch interessieren nicht die Geschwindigkeiten zum

Halbschritt  $t + \Delta t/2$  sondern zum Zeitpunkt  $t + \Delta t$ . Beim Velocity Verlet Algorithmus wählt man wie beim Euler Algorithmus (Gleichungen (2.1) und (2.2)) jeweils eine Taylorentwicklung für die Bestimmung der Position und der Geschwindigkeit. Anstatt bei der Berechnung der Geschwindigkeit jedoch die Beschleunigung zum Zeitpunkt  $t$  einfließen zu lassen, wird der Mittelwert der Beschleunigung zu den Zeitpunkten  $t$  und  $t + \Delta t$  verwendet. Die Iterationsvorschrift lautet also:

$$\begin{aligned}\vec{r}(t + \Delta t) &= \vec{r}(t) + \vec{v}(t)\Delta t + \frac{1}{2m}\vec{F}(t)\Delta t^2 \\ \vec{v}(t + \Delta t) &= \vec{v}(t) + \frac{1}{m} \left[ \frac{\vec{F}(t) + \vec{F}(t + \Delta t)}{2} \right] \Delta t\end{aligned}$$

Realisiert wird dieses Schema, indem aus der aktuellen Kraft die Geschwindigkeit im nächsten Halbschritt berechnet wird (Gleichung (2.6)). Aus der Geschwindigkeit zum Halbschritt wird die Position zum Zeitpunkt  $t + \Delta t$  ermittelt (Gleichung (2.7)). Zuletzt wird die Kraft zum Zeitpunkt  $t + \Delta t$  bestimmt (Gleichung (2.8)) und hieraus die Geschwindigkeit zum selben Zeitpunkt (Gleichung (2.9)).

$$\vec{v}(t + \Delta t/2) = \vec{v}(t) + \frac{\Delta t}{2m}\vec{F}(t) \quad (2.6)$$

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t + \Delta t/2)\Delta t \quad (2.7)$$

$$\vec{F}(t + \Delta t) = -\nabla U(\vec{r}(t + \Delta t)) \quad (2.8)$$

$$\vec{v}(t + \Delta t) = \vec{v}(t + \Delta t/2) + \frac{\Delta t}{2m}\vec{F}(t + \Delta t) \quad (2.9)$$

### Andere Integratoren

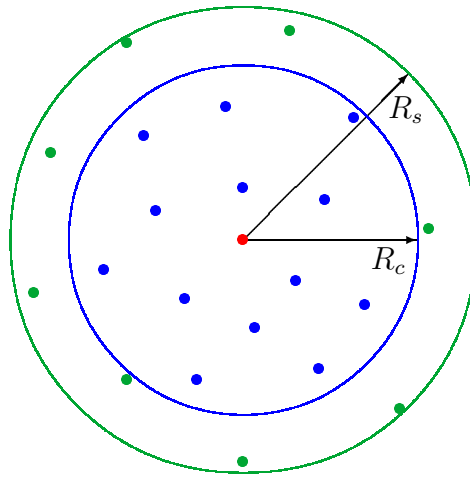
Neben den bereits diskutierten Integratoren gibt es noch unter anderen die besondere Klasse der Predictor-Corrector Methoden. Hier wird zunächst nur eine Vorhersage für die neuen Positionen gemacht, welche dann für die Kraftberechnung verwendet wird. Im Anschluß daran wird die Vorhersage leicht korrigiert.

## 2.6 Nachbarschaftslisten

Will man die Wechselwirkungen auf die Teilchen berechnen, so muss man in jedem Zeitschritt  $N(N-1)/2$  potentielle Wechselwirkungspartner betrachten. Dies stellt einen enorm großen Aufwand dar. Vor allem bei kurzreichweitigen Wechselwirkungen wird ein Teilchen von nur wenigen Teilchen aus der unmittelbaren Umgebung beeinflusst. Der Einfluß, der von entfernteren Teilchen stammt, ist so gering, dass er vernachlässigt werden kann. Es wird daher ein Cutoff-Radius  $r_c$  eingeführt. Dies bedeutet, dass nur Teilchen, deren Abstand voneinander kleiner ist als der Cutoff-Radius, miteinander wechselwirken. Bei langreichweitigen Wechselwirkungen ist diese Vorgehensweise sehr problematisch, da selbst entfernte Teilchen einen

signifikanten Einfluss aufeinander ausüben können. Es sind daher spezielle Korrekturmechanismen anzuwenden.

Realisiert werden kann diese Vorgehensweise mit Hilfe von Verlet-Listen. Für jedes Teilchen werden die entsprechenden Nachbarpartikel in einer Liste abgespeichert. Damit nicht im Verlauf der Simulation unbemerkt zwei Teilchen zu Nachbarn werden können, wird zusätzlich noch ein Skinradius  $r_s > r_c$  eingeführt. Damit zwei Teilchen  $i$  und  $j$  zu Nachbarn werden können, muss sich Teilchen  $i$  zuvor innerhalb des Skinradius um Teilchen  $j$  befunden haben (und umgekehrt). Diese Listen müssen regelmäßig aktualisiert werden [5].



**Abbildung 2.3:** Das rote Teilchen wechselwirkt mit allen Teilchen (blau), deren Abstand kleiner gleich dem Cutoff-Radius  $R_c$  ist. Innerhalb des Skinradius  $R_s$  befinden sich die Teilchen (grün), deren Abstand im nächsten Zeitschritt kleiner gleich dem Cutoff-Radius  $R_c$  werden können. Sie sind somit mögliche Wechselwirkungspartner im nächsten Zeitschritt.

Die Methode der Verlet-Listen hat den frei wählbaren Parameter Skinradius. Wird dieser klein gewählt, so sind viele Updates der Nachbarschaftslisten nötig, ist er groß, so gibt es wenige Updates, aber viele Abfragen für die Kraftberechnung, ob nun zwei Teilchen miteinander wechselwirken.

Eine andere Methode, um den Aufwand zu reduzieren, stellen die Linked-Cell-Listen dar. Ziel ist, den Listenaufbau durch lokale Operationen zu realisieren. Das Gesamtsystem wird hierbei in kubische Zellen untergliedert, deren Kantenlängen dem Cutoff-Radius entsprechen. Für jede Zelle werden einmal zu Beginn der Simulation die Nachbarzellen bestimmt. Neben den Nachbarzellen werden auch die Teilchen, die sich in der Zelle befinden, abgespeichert. Ein Teilchen aus Zelle  $i$  kann daher nur mit Teilchen interagieren, die in derselben Zelle oder in einer der Nachbarzellen liegen. Dieser Vorgehensweise liegen lokale Operationen zu Grun-

de. Da man mit kubischen Zellen die Cutoff-Kugel jedoch nicht nachbilden kann, muss bei der Berechnung der Kräfte überprüft werden, ob zwei Teilchen einen geringeren Abstand zueinander haben als den Cutoff-Radius, d.h. ob sie mit einander wechselwirken. Hierbei entsteht ein Mehraufwand, den man gerne vermeiden möchte.

Durch eine Kombination der beiden Verfahren erhält man ein Verfahren mit der Lokalität von Linked-Cell-Listen und der Effizienz von Verlet-Listen (z.B. der Hierarchischer Algorithmus).



# Kapitel 3

## Das Verfahren

Particle-Particle Methoden, kurz PP-Methoden genannt, wie z.B. der eingangs im Kapitel 1.1 vorgestellte naive Lösungsansatz, sind zwar sehr genau, haben aber einen extrem hohen Rechen- und Zeitaufwand. Particle-Mesh Methoden (PM-Methoden genannt), wie z.B. die im Abschnitt 1.2 kurz beschriebene, sind zwar sehr schnell, in der Regel allerdings sehr ungenau. Wünschenswert ist daher eine Kombination aus beiden, die die jeweiligen Nachteile der Simulationsarten kompensiert, die jeweiligen Vorzüge jedoch erhält.

Die hier verwendete Kombination sieht kurz umrissen folgendermaßen aus: Das gesamte Simulationsgebiet wird in viele kleinere Gitterzellen untergliedert. Das Potential  $\Phi_i^{(near)}$  auf ein Teilchen  $i$ , welches von Wechselwirkungen von Teilchen aus dem Nahfeld, d.h. aus derselben Gitterzelle und den Nachbarzellen, stammt, wird mittels einer PP-Methode ermittelt, das Potential  $\Phi_i^{(far)}$  aus dem Fernfeld, welches von den Wechselwirkungen aller anderen Teilchen der Simulation herrührt, mit Hilfe einer PM-Methode. Um das Gesamtpotential  $\Phi_i$  auf ein Teilchen  $i$  der Simulation zu berechnen, müssen lediglich  $\Phi_i^{(near)}$  und  $\Phi_i^{(far)}$  addiert werden. Die Gesamtkraft  $\vec{F}_i$ , die auf das Teilchen  $i$  einwirkt, berechnet sich analog als Summe der Kräfte  $\vec{F}_i^{(near)}$  aus dem Nahfeld und der Kräfte  $\vec{F}_i^{(far)}$  aus dem Fernfeld.

Um eine verständliche Beschreibung der Methode zu gewährleisten, ist es an dieser Stelle sinnvoll, einige Vereinbarungen von Bezeichnungen vorzunehmen. Im Folgenden wird mit

- $C_i$  Gebiet der Nachbarzellen, in dem sich Teilchen  $i$  befindet
- $P_i$  Gitterpunkte der Zelle, in der sich Teilchen  $i$  befindet
- $G_i$  Gitterzellen, die den Gitterpunkt  $i$  als Eckpunkt haben

bezeichnet.

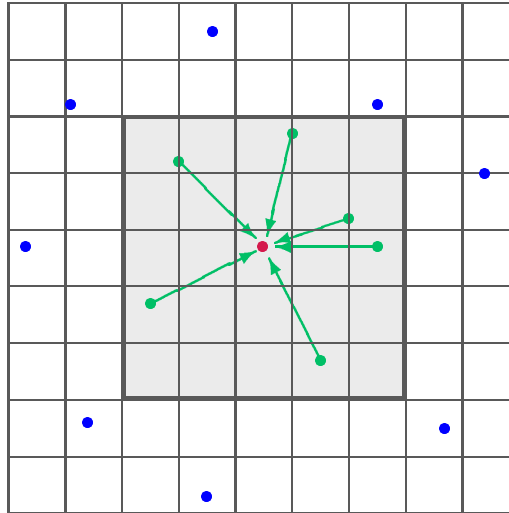
### 3.1 Nahfeldberechnung mittels PP-Methode

Um den Anteil des Potentials auf ein Teilchen  $i$  nach der Particle-Particle Methode zu berechnen, werden die Wechselwirkungen von allen anderen Teilchen aus den Nachbarzellen auf das Teilchen  $i$  ermittelt [2]. Mathematisch bedeutet dies:

$$\Phi_i^{(near)} = \sum_{j \in C_i, j \neq i} \frac{q_j}{|\vec{r}_i - \vec{r}_j|}$$

Die Anzahl der Nachbarzellen kann unterschiedlich groß gewählt werden. Das kleinstmögliche “Gebiet” der Nachbarzellen umfasst lediglich eine Zelle - nämlich die Zelle, in der sich Teilchen  $i$  befindet. Die nächst größere Anzahl an Nachbarzellen ist im dreidimensionalen Fall 27: die eigene Zelle und alle anderen Zellen, die mindestens einen Punkt mit der eigenen Zelle gemein haben. Es werden also zum nächst kleineren Gebiet von Nachbarzellen alle umliegenden Zellen mit direktem Kontakt zu diesem Gebiet hinzugenommen. Dies bedeutet, dass die nächst größere Nachbarzellenanzahl 125 ist.

Die Vorgehensweise der PP-Methode ist zur Verdeutlichung in der Abbildung 3.1 für den zweidimensionalen Fall schematisch dargestellt.



**Abbildung 3.1:** Schematische Darstellung der PP-Methode im 2D: Zur Berechnung des Potentials auf ein Teilchen  $i$  (rot) werden im PP-Teil des Algorithmus’ nur die Teilchen aus den Nachbarzellen (grau unterlegter Bereich) berücksichtigt.

Durch die Verwendung einer PP-Methode für den Nahbereich wird gewährleistet, dass der größte, sich schnell ändernde Anteil am Gesamtpotential stets exakt berechnet wird. Der Aufwand für diesen Teil des Algorithmus’ ist proportional zu  $N_{PP}^2 - N_{PP}$ , wobei  $N_{PP}$  die Anzahl der Teilchen ist, welche sich durchschnittlich



in den Nachbarzellen befinden.

Für die Integration der Bewegungsgleichungen sind jedoch nicht die Potentiale sondern die Kräfte relevant, die sich aus dem negativen Gradienten der potentiellen Energie ergeben. Die Kräfte aus dem Nahfeld berechnen sich durch

$$\vec{F}_i^{(near)} = q_i \sum_{j \in C_i, j \neq i} \frac{q_j}{|\vec{r}_i - \vec{r}_j|^3} (\vec{r}_i - \vec{r}_j).$$

## 3.2 Fernfeldberechnung mittels PM-Methode

Bei der PM-Methode wird die Ladung eines jeden Teilchens auf die Eckgitterpunkte der Zelle interpoliert, in der sich das jeweilige Teilchen gerade befindet:

$$\tilde{q}_h^i = \omega_h^i \cdot q_i$$

Hierbei ist  $\tilde{q}_h^i$  der Teil der Ladung von Teilchen  $i$ , der dem Gitterpunkt  $h$  zugeordnet wird. Auf das Gewicht  $\omega_h^i$ , welches durch die Interpolation bestimmt wird, wird in Kapitel 3.2.1 näher eingegangen.

Da es sich bei der Interpolation um eine Aufteilung der Ladung in Teilladungen handelt, die dann auf die Gitterpunkte aufgetragen werden, ist die Summe dieser Teilladungen gleich der Ladung des Teilchens.

$$q_i = \sum_{h \in P_i} \tilde{q}_h^i$$

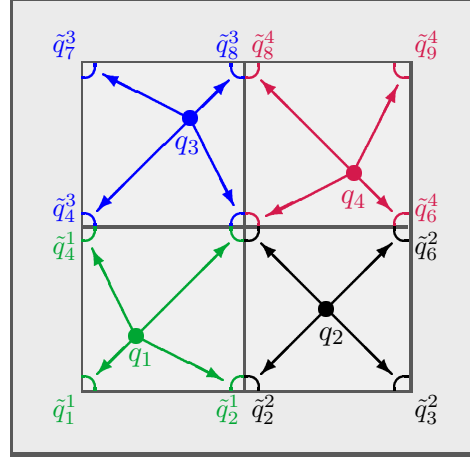
Weil ein Gitterpunkt Eckpunkt von mehreren Zellen ist (im Dreidimensionalen sind es 8) und in jeder Gitterzelle mehrere Teilchen sind, deren Ladung auf die umliegenden Gitterpunkte verteilt werden, ist die Gesamtladung eines Gitterpunktes gleich der Summe der einzelnen Teilladungen, die dem Gitterpunkt zugewiesen werden:

$$\tilde{q}_h = \sum_{i \in G_h} \omega_h^i \cdot q_i = \sum_{i \in G_h} \tilde{q}_h^i$$

In Abbildung 3.2 werden diese Tatsachen beispielhaft für einen zweidimensionalen Fall dargestellt.

Um den Anteil des Potentials aus der PM-Methode bestimmen zu können, werden die Wechselwirkungen zwischen den Eckgitterpunkten der Zelle, in der das Teilchen  $i$  liegt, und den übrigen Gitterpunkten ermittelt. Kompakt wird dies durch eine Matrix-Vektor-Multiplikation (MV-Multiplikation) ausgedrückt [6, 7, 8], wobei in der Matrix  $\tilde{A}$  die inversen Abstände der Gitterpunkte zueinander und in dem Vektor  $\tilde{q}$  die zeitabhängigen Ladungen der Gitterpunkte stehen.

$$\tilde{\Phi} = \tilde{A} \cdot \tilde{q}$$



**Abbildung 3.2:** Die Ladungen aller Teilchen werden auf die umliegenden Gitterpunkte interpoliert. Der innere Gitterpunkt erhält Potentialanteile von Teilchen aus allen vier Zellen.

Da die Ladung, die mit der PP-Methode bereits berechnet wurde, nicht mehr in die Berechnung eingehen darf, sind nur die Gitterpunktladungen zu berücksichtigen, die nicht von Teilchen aus  $C_i$  stammen. Bei der Verwendung der MV-Multiplikation muss daher an dieser Stelle eine Korrektur erfolgen.

$$\tilde{\Phi}_h^{(cor)} = \tilde{\Phi}_h - \sum_{g \in C_i} \tilde{A}_{h,g} \tilde{q}_g = \tilde{\Phi}_h - \sum_{g \in C_i} \frac{\tilde{q}_g}{|\vec{r}_h - \vec{r}_g|}$$

Zu beachten ist, dass für jede Gitterzelle, die den Gitterpunkt  $h$  als Eckpunkt hat, unterschiedliche Gitterzellen zum Nahfeld gehören, so dass die Korrektur nur bezogen auf die jeweils betrachtete Zelle Gültigkeit besitzt. Es handelt sich somit nicht um eine globale Korrektur auf den Gitterpunkten sondern um eine lokale.

Die korrigierten Potentiale auf den Eckgitterpunkten müssen nun lediglich auf das Teilchen zurückinterpoliert werden [2]. Damit gilt für den Potentialanteil aus der PM-Methode auf ein Teilchen  $i$ :

$$\Phi_i^{(far)} = \sum_{h \in P_i} \omega_h^i \cdot \tilde{\Phi}_h^{(cor)}$$

Aus dem korrigierten Potentialfeld, welches auf den Gitterpunkten durch  $\tilde{\Phi}^{(cor)}$  gegeben ist, wird durch ein finites Differenzenverfahren das elektrische Feld  $\vec{E}$  berechnet [2], welches auf die Teilchen zurückinterpoliert wird und worüber die Kräfte auf die Teilchen bestimmt werden.

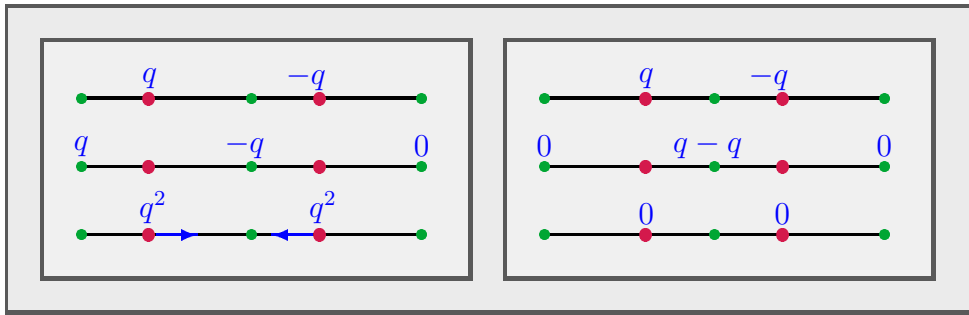
$$\vec{F}_i^{(far)} = q_i \sum_{h \in P_i} \omega_h^i \vec{E}_h = -q_i \sum_{h \in P_i} \omega_h^i \vec{D}_h^{(n)} \tilde{\Phi}_h^{(cor)}$$

Hierbei ist  $\vec{D}_h^{(n)}$  ein finiter Differenzen-Operator des Gradienten bezüglich des Gitterpunktes  $h$  und einer Ordnung  $n$ .

### 3.2.1 Gewichtungsfunktion $\omega_h^i$

Die Gewichtungsfunktion  $\omega_h^i$  legt fest, wie groß der Ladungsanteil von Teilchen  $i$  ist, welchen der Gitterpunkt  $h$  erhält. Es gibt sehr viele Möglichkeiten für eine Gewichtungsfunktion. Für eine gute Gewichtungsfunktion gilt jedoch, dass bei Anwendung der Funktion das Potential, welches von der Eigenladung eines Teilchens herrührt, nicht einfließt, und dass sich durch kleine Positionsänderungen eines Teilchens das Gesamtpotential nur geringfügig ändert.

Eine bekannte Methode ist das NGP-Schema (**N**earest **G**rid **P**oint). Hierbei wird die gesamte Ladung von einem Teilchen  $i$  dem nächst gelegenen Gitterpunkt  $h$  zugeordnet. Dieses Schema ist sehr einfach, allerdings ist die Anwendung dieser Methode nicht ratsam, da durch eine kleine Änderung der Position eines Teilchens ein anderer Gitterpunkt zum nächstliegenden Gitterpunkt werden kann und dadurch extreme Sprünge im Potential entstehen können [2]. In Abbildung 3.3 wird dies verdeutlicht.

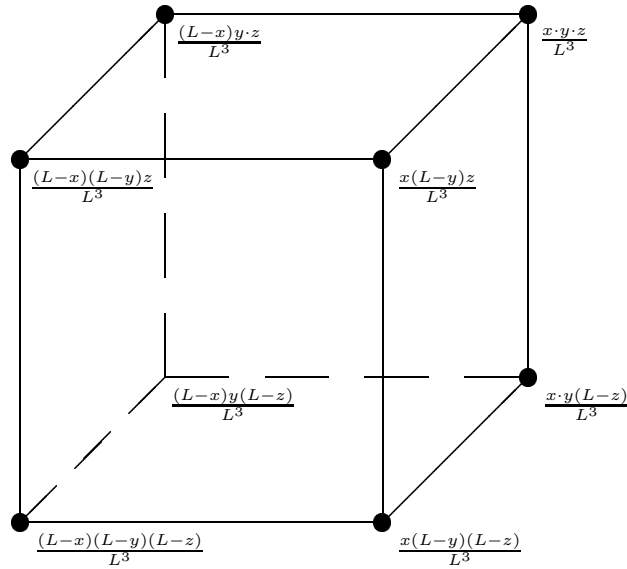


**Abbildung 3.3:** Instabilität des NGP-Schemas vereinfacht dargestellt für den eindimensionalen Fall: In beiden Teilen der Grafik sind jeweils drei Gitterpunkte (grün) und zwei Ladungspunkte (rot) mit den Ladungen  $q$  und  $-q$ . Der rechte Ladungspunkt liegt in beiden Fällen dem mittleren Gitterpunkt am nächsten: Dieser erhält die Ladung von diesem Ladungspunkt. Der linke Ladungspunkt liegt in der linken Teilgrafik dem linken Gitterpunkt am nächsten, so dass dieser die gesamte Ladung  $q$  erhält. In der rechten Teilgrafik dagegen ist der Ladungspunkt etwas nach rechts verschoben, so dass der nächstliegende Gitterpunkt der mittlere ist und jenem die Ladung zugeordnet wird. Obwohl nur eine geringe Änderung der Eingangsdaten vorgenommen wurde, ist das errechnete Potential und somit auch die berechnete Kraft in den beiden Fällen sehr unterschiedlich (3. Reihe).

Etwas aufwendiger, dafür stabil, ist das CIC-Schema (**C**loud **I**n **C**ell). Hierbei wird allen Gitterpunkten der Zelle, in welcher das jeweilige Teilchen liegt, ein Ladungsanteil zugeordnet [2]. Bei Abbildung 3.2 ist dieses Schema bereits angewendet worden. Die Größe des Ladungsanteils, den ein Gitterpunkt erhält, hängt

in der Regel von der Lage des Gitterpunktes zu dem Ladungspunkt ab (relativ gesehen zu der Lage der übrigen Gitterpunkte der Gitterzelle).

Im eindimensionalen Fall bedeutet dies eine Aufteilung der Ladung entsprechend des inversen Abstandes zu den Gitterpunkten, im zweidimensionalen entsprechend der inversen Flächeninhalte und im dreidimensionalen Fall entsprechend des inversen Volumeninhaltes.



**Abbildung 3.4:** CIC-Schema im dreidimensionalen Fall: Die Gitterpunkte mit den zugeordneten Ladungsanteilen einer Punktladung mit den Koordinaten  $(x, y, z)$  innerhalb der Gitterzelle der Kantenlänge  $L$

In Abbildung 3.4 sind für den dreidimensionalen Fall die Ladungsanteile den jeweiligen Gitterpunkten zugeordnet. Die Variablen  $x$ ,  $y$  und  $z$  sind hierbei die Koordinaten eines Ladungspunktes aus der Gitterzelle,  $L$  ist die Kantenlänge der kubischen Gitterzelle. Für ein zweidimensionales bzw. eindimensionales Problem kann man die benötigten Ladungsanteile ebenfalls aus der Abbildung 3.4 ablesen. Im zweidimensionalen Fall gilt  $z = 0$ . Damit bekommen nur die Gitterpunkte der Standfläche einen Ladungsanteil zugeordnet. Im eindimensionalen Fall gilt zusätzlich  $y = 0$ , und nur die beiden vorderen Gitterpunkte der Standfläche erhalten einen Ladungsanteil.

### 3.2.2 Einsatz der Wavelets im Verfahren

Zusammenfassend besteht die PM-Methode, wie gezeigt, aus den folgenden drei Schritten:

1. Ladung der Teilchen auf die Gitterpunkte interpolieren
2. Wechselwirkungen zwischen den Gitterpunkten berechnen
3. Potential bzw. Kräfte von den Gitterpunkten auf die Teilchen interpolieren

Schritt 1 und 3 haben eine Komplexität proportional zu  $N$ , der Gesamtanzahl der Teilchen in der Simulation. Die Ordnung von Schritt 2 hängt von der Anzahl der Gitterpunkte  $N_G$  ab, welche durch die Teilchenanzahl  $N$  und die durchschnittliche Anzahl von Teilchen je Gitterzelle festgelegt ist. Insgesamt ergibt sich eine Komplexität von  $\alpha N + \beta N_G^2$ , wobei  $N_G \ll N$ . Dies bedeutet, dass das Verfahren immer noch eine quadratische Komplexität aufweist, lediglich der Vorfaktor ist verkleinert worden.

Der Aufwand der Matrix-Vektor-Multiplikation sinkt wesentlich, wenn zu einer Darstellung übergegangen wird, bei der die Matrix dünn besetzt, d.h. komprimiert, ist. Ein solcher Übergang wird mit Hilfe einer geeigneten Transformation durchgeführt. In dem hier verwendeten Verfahren wurde die Wavelet-Transformation implementiert, welche im Kapitel 4 eingeführt und ausführlich diskutiert wird.

Die Kompression wird in der Regel nicht von der Transformation selber erreicht sondern durch ein Threshold-Verfahren, mit dem die Transformation oftmals gekoppelt wird. Bei einem Threshold-Verfahren werden Werte, deren Betrag kleiner als der Schwellenwert (Threshold) ist, zu Null gesetzt, Werte dem Betrag nach oberhalb des Schwellenwertes werden nach bestimmten Regeln behandelt, z.B. unverändert gelassen (Hard-Thresholding) oder dem Betrag nach um den Schwellenwert verkleinert (Soft-Thresholding). Auf das Thresholding wird in Kapitel 4.7 näher eingegangen.

Das bisher vorgestellte Verfahren wird dahingehend modifiziert, dass die Matrix-Vektor-Multiplikation im Wavelet-Raum durchgeführt wird. Hierzu wird die Matrix  $\tilde{A}$  durch Anwendung der Wavelet-Transformation, die mit einem Threshold-Verfahren gekoppelt ist, in eine dünn besetzte Matrix  $\hat{A}_t$  umgewandelt. Abbildungen zu den verschiedenen Darstellungen der Abstandsmatrix sind im Anhang B zu finden.

Da sich die Abstände der Gitterpunkte zueinander und somit die Einträge der Matrix  $\tilde{A}$  im Verlauf der Simulation niemals ändern, genügt es, die Transformation einmal zu Beginn der Simulation durchzuführen. Der Vektor  $\tilde{q}$  beinhaltet die zeitabhängigen Ladungen auf den Gitterpunkten. Somit verändert er sich in jedem Zeitschritt und muss stets vor der Durchführung der Matrix-Vektor-Multiplikation transformiert werden.

$$\hat{\Phi} = \hat{A}_t \cdot \hat{q}$$

Durch Rücktransformation von  $\hat{\Phi}$  ist das globale Potential  $\tilde{\Phi}$  auf den Gitterpunkten gegeben. Aufgrund der zuvor beschriebenen Modifikationen wird es nötig, die Nahfeldkorrektur anzupassen.

$$\tilde{\Phi}_h^{(cor)} = \tilde{\Phi}_h - \sum_{g \in C_i} \tilde{A}_{t,h,g} \tilde{q}_g$$

Die Matrix  $\tilde{A}_t$  ist hierbei die Rücktransformation von  $\hat{A}_t$ , von der komprimierten wavelettransformierten Abstandsmatrix. Aufgrund dieser Abänderungen wird der Aufwand des Verfahrens auf eine lineare Komplexitätsordnung verringert, wie die Ergebnisse in Kapitel 5.4 belegen.

# Kapitel 4

## Wavelets

### 4.1 Einführung

Bei der Wavelet-Transformation (WT) geht es wie bei jeder Transformation darum, eine Funktion  $f$  als Linearkombination von Grundfunktionen  $\psi_k$  und entsprechenden Koeffizienten  $d_k$  darzustellen. Es handelt sich also um eine lineare Abbildung zwischen gewichteten  $L^2$ -Räumen.

$$f = \sum_{k \in \mathbb{N}} d_k \psi_k$$

Eine gute Transformation liegt vor, wenn mit möglichst wenigen Koeffizienten bei geringem Aufwand eine bestmögliche Approximation der Funktion  $f$  erreicht wird. Die Wavelet-Transformation hat viele Gemeinsamkeiten mit der Fourier-Transformation (FT). Beide Transformationen sind Frequenz-Transformationen, d.h. sie bilden Funktionen eines Funktionenraumes auf die Koeffizienten von Basisfunktionen ab, wobei die Basisfunktionen eine Lokalität im Frequenzspektrum aufweisen. Im Falle der FT wird die Ausgangsfunktion in ihre Sinus- und Cosinus-Bestandteile mit verschiedenen Frequenzen zerlegt. Bei der WT dagegen werden als Basisfunktionen sogenannte “Wavelets” verwendet.

Im Gegensatz zum Sinus und Cosinus besitzt ein Wavelet nicht nur Lokalität im Frequenzspektrum sondern auch im Zeitbereich [20]. Wavelets haben daher die Form von nach außen hin abklingenden Wellen, daher der Name Wavelet – “Wellchen”.

Diese Lokalitäten sollen am folgenden Beispiel näher erläutert werden. Wird ein Akkord von drei Tönen untersucht, so erhält man sowohl durch die FT als auch durch die WT die zugehörigen Frequenzen dieser Töne. Werden diese Töne nun einzeln nacheinander gespielt, so liefert die FT wiederum die Frequenzen dieser Töne, die Information, wann welche Frequenz auftrat ist aus der Fourier-Transformierten nicht erkenntlich. Bei Verwendung der WT dagegen sind aus dem transformierten Signal nicht nur die Frequenzen der Töne erkennbar, sondern auch

der Zeitpunkt an bzw. die Zeitdauer in der sie auftraten.

Den Nachteil der FT, keine Zeitlokalität aufzuweisen, versucht die gefenster- te Fourier-Transformation (WFT) auszugleichen [10, 16]. Hierbei wird das Signal nicht in seiner Gesamtheit untersucht, sondern nur in einem bestimmten Intervall. Dieses Zeitfenster wird dann sukzessive über das gesamte Signal verschoben. Es liegt allerdings in Abhängigkeit der Intervallbreite entweder eine gute Zeit- oder eine gute Frequenzauflösung vor. Des Weiteren ist diese Methode numerisch sehr aufwendig.

Wavelets sind im Allgemeinen eher kompliziert aufgebaut. Obwohl es eine un- endlich große Anzahl von Wavelets gibt, sind nur sehr wenige im praktischen Gebrauch.

Der Vorteil der Wavelet-Transformation gegenüber der Fourier-Transforma- tion ist zum einen die zeitliche Lokalität der Basisfunktionen (Wavelets), zum anderen die geringe Komplexität der Transformation von  $O(N)$  im Gegensatz zu  $O(N \log N)$  bei der Fourier-Transformation, wobei  $N$  die Anzahl der Abtastzeit- punkte ist.

Die Wavelet-Transformation wird in drei Gruppen eingeteilt: Die kontinuier- liche Wavelet-Transformation (CWT), welche auf kontinuierliche Funktionen an- wendbar ist (vgl. 4.2), die diskrete Wavelet-Transformation (DWT) für abgetaste- te Funktionen und die schnelle Wavelet-Transformation (FWT), die eine schnelle und stabile numerische Umsetzung der diskreten Wavelet-Transformation ist.

## 4.2 Kontinuierliche Wavelet-Transformation

Eine Funktion  $\psi \in L^2(\mathbb{R})$  mit ihrer Fourier-Transformierten  $\hat{\psi}(\omega)$ , welche die Zulässigkeitsbedingung

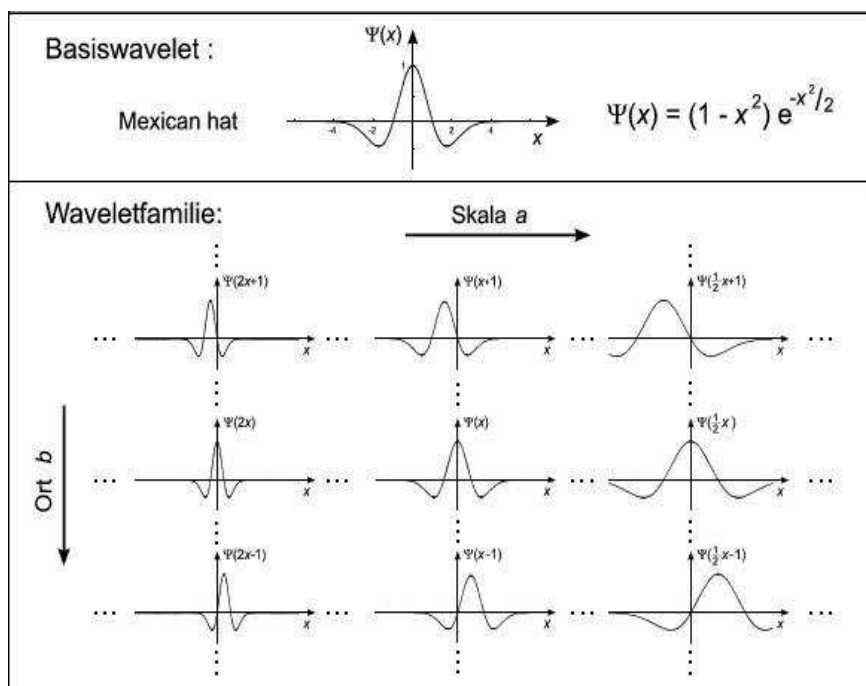
$$0 < c_\psi := 2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < +\infty \quad (4.1)$$

erfüllt, heißt Wavelet. Diese Bedingung kann nur erfüllt werden, wenn die Fourier- Transformierte von  $\psi$  an der Stelle  $\omega = 0$  eine Nullstelle besitzt. Hieraus folgt, dass der Mittelwert von  $\psi$  verschwindet.

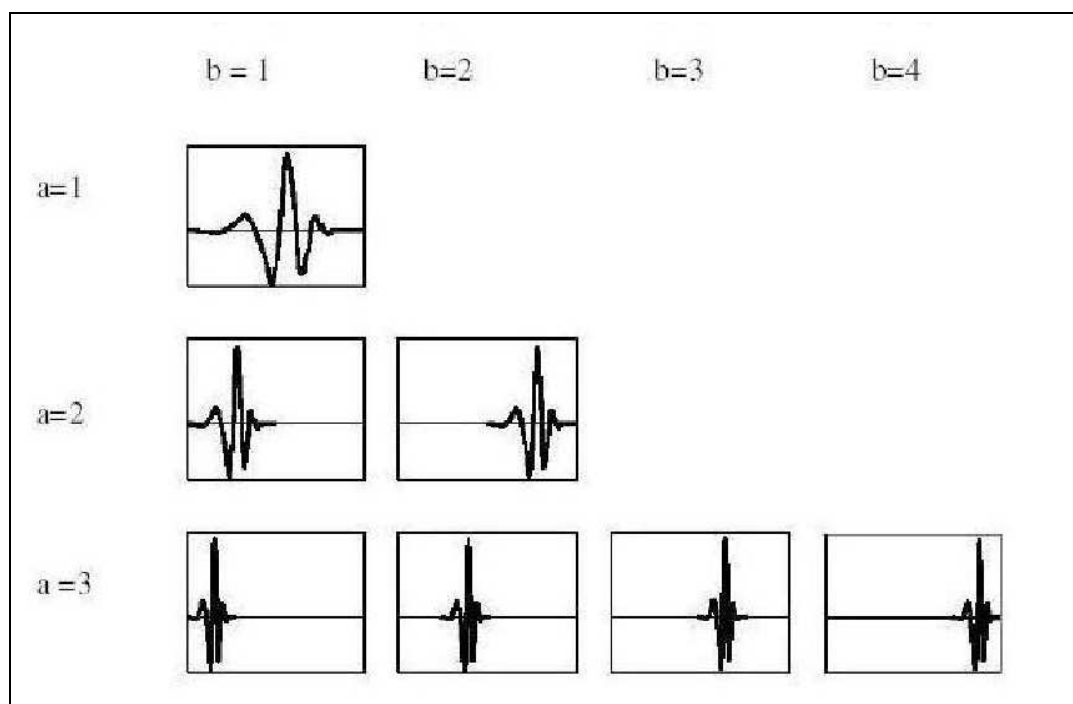
$$0 = \hat{\psi}(0) = (2\pi)^{-1/2} \int_{-\infty}^{\infty} \psi(t) \exp(-it \cdot 0) dt = \int_{-\infty}^{\infty} \psi(t) dt \quad (4.2)$$

Allgemein wird bei einer CWT folgendermaßen vorgegangen: Gewählt wird ein Basis-Wavelet  $\psi$ , auch “Mutterwavelet” genannt. Dieses wird skaliert (gestreckt





**Abbildung 4.1:** Verschiedene Translationen und Dilatationen eines Wavelets am Beispiel des Wavelets “Mexikanerhut”.



**Abbildung 4.2:** Verschiedene Translationen und Dilatationen eines Wavelets am Beispiel eines *Daub4*-Wavelets .

oder gestaucht) um den Faktor  $a$  und verschoben durch den Verschiebungsfaktor  $b$  [16].

$$\psi\left(\frac{t-b}{a}\right) \quad a, b \in \mathbb{R}, a \neq 0$$

Diese durch Dilatation und Translation entstandenen Varianten des Mutterwavelets nennt man auch “Kinder” des Wavelets. Die Abbildungen 4.1 und 4.2 sind Beispiele für ein “Mutterwavelet” und seine “Kinder”.

Um die Koeffizienten der Wavelet-Analyse eines Signals zu erhalten, wird das Signal mit dem Mutterwavelet gefaltet, anschließend mit seinen “Kindern”. So erhält man Signale, welche immer noch zeit- bzw. ortsabhängig sind, aber gleichzeitig Informationen über die Frequenz enthalten. Das Signal  $f(t)$  wird also in eine Funktion abhängig von der Skala  $a$  und der Zeit  $b$  überführt [16, 20].

$$(L_\psi f(t))(a, b) = \frac{1}{\sqrt{c_\psi}} \cdot |a|^{-1/2} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt \quad a, b \in \mathbb{R}, a \neq 0 \quad (4.3)$$

Als Isometrie zwischen Hilbert-Räumen wird die Wavelet-Transformation auf ihren Bildbereich durch ihre adjungierte Abbildung invertiert. Eine Funktion  $f \in L^2(\mathbb{R})$  besitzt daher die Darstellung

$$f(x) = \frac{1}{\sqrt{c_\psi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (L_\psi f(t))(a, b) \frac{1}{\sqrt{|a|}} \cdot \psi\left(\frac{x-b}{a}\right) \frac{dad b}{a^2} \quad (4.4)$$

mit der Wavelet-Transformation  $L_\psi$  zum Wavelet  $\psi$  [19, 20]. Diese Darstellung ist, wie bereits erwähnt, für kontinuierliche Funktionen anwendbar. Bei dem in dieser Arbeit betrachteten Problem sind die Zielfunktionen (Ladungsvektoren und Abstandsmatrix) jedoch diskret, so dass die kontinuierliche Wavelet-Transformation nicht weiter betrachtet wird.

## 4.3 Diskrete Wavelet-Transformation

Bei der diskreten Wavelet-Transformation werden die Wavelets nur um ganze Zahlen gestaucht und verschoben. Meistens wird um den dyadischen Faktor 2 skaliert.

$$\psi(2^k t + l) \quad k, l \in \mathbb{Z}$$

Die Multiskalen-Analyse (MSA) ist eine mathematische Methode, die eine schnelle und stabile Wavelet-Analyse (Zerlegung, Transformation) und -Synthese (Rekonstruktion) erlaubt und somit Grundlage der diskreten Wavelet-Transformation ist [7, 8, 19, 20].

### 4.3.1 Multiskalen-Analyse

Das Signal  $f$  soll in seine hoch- und niederfrequenten Anteile aufgespalten werden. Hierzu sei  $f \in V_{-1} \subset L^2(\mathbb{R})$ , wobei  $V_{-1}$  ein beliebiger Unterraum des  $L^2(\mathbb{R})$  ist. Die “glatten”, niederfrequenten Anteile von  $f$  werden durch die Projektion  $P_0 f$  auf den Unterraum  $V_0$  beschrieben, der die “glatten” Funktionen aus  $V_{-1}$  enthält. Der Raum  $W_0$ , welcher die hochfrequenten Anteile beschreibt, ist das orthogonale Komplement von  $V_0$  in  $V_{-1}$ . Die Projektion von  $f$  auf  $W_0$  wird mit  $Q_0 f$  bezeichnet. Somit wird das Signal  $f \in V_{-1} \subset L^2(\mathbb{R})$  durch

$$f = P_0 f + Q_0 f \quad \text{mit} \quad f \in V_{-1} = V_0 \oplus W_0$$

dargestellt.

Der niederfrequente Anteil  $P_0 f$  wird nun auf dieselbe Weise zerlegt, indem der Raum  $V_0$  wiederum als Summe der orthogonalen Unterräume  $V_1$ , der die niederfrequenten, und  $W_1$ , der die hochfrequenten Elemente von  $V_0$  enthält, geschrieben wird. Die zugehörigen Projektoren sind  $P_1$  und  $Q_1$ .

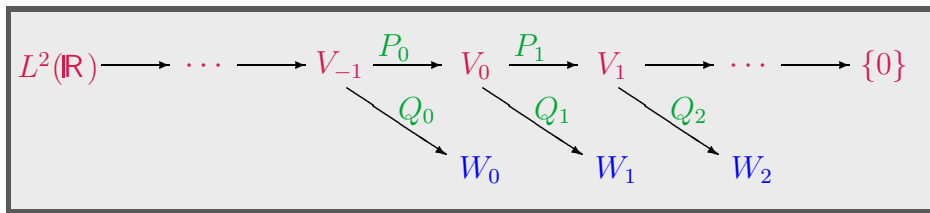
$$P_0 f = P_1 f + Q_1 f \quad \text{mit} \quad P_0 f \in V_0 = V_1 \oplus W_1$$

Durch Fortsetzen dieses Verfahrens entsteht das folgende Rekursionsschema, welches auch in Abbildung 4.3 schematisch dargestellt ist [7, 8, 19, 20].

$$P_m f = P_{m+1} f + Q_{m+1} f \quad V_m = V_{m+1} \oplus W_{m+1} \quad \forall m \in \mathbb{Z}$$

Die Funktion  $f$  kann nun bei einer Zerlegung bis zum Level  $M$  wie folgt dargestellt werden:

$$f = P_M f + \sum_{k=0}^M Q_k f$$



**Abbildung 4.3:** Schematische Darstellung einer Multiskalenanalyse des  $L^2(\mathbb{R})$ :  $P_i f$ ,  $i \in \mathbb{Z}$  enthält die Details von  $f$  nach der  $i$ -ten Zerlegung,  $Q_i f$ ,  $i \in \mathbb{Z}$  beinhalten die Anteile von  $f$  zu einem bestimmten Frequenzband.

Mathematisch ist eine Multiskalenanalyse des  $L^2(\mathbb{R})$  folglich eine aufsteigende Folge von abgeschlossenen Unterräumen  $V_m \subset L^2(\mathbb{R})$ ,  $m \in \mathbb{Z}$ ,

$$\{0\} \subset \cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset \cdots \subset L^2(\mathbb{R}),$$

so dass gilt

$$\overline{\bigcup_{m \in \mathbb{Z}} V_m} = L^2(\mathbb{R}) \quad (4.5)$$

$$\bigcap_{m \in \mathbb{Z}} V_m = \{0\} \quad (4.6)$$

$$f(\cdot) \in V_m \Leftrightarrow f(2^m \cdot) \in V_0 \quad (4.7)$$

Die erste Gleichung besagt, dass der  $L^2$ -Abschluss der Vereinigungen der Räume  $V_m$ ,  $m \in \mathbb{Z}$ , gleich dem Raum  $L^2(\mathbb{R})$  ist, wodurch die Vollständigkeit gegeben ist. Die zweite Gleichung legt den Ausschluss von Redundanz fest, da die einzige Funktion, die in allen Räumen  $V_m$  enthalten ist, die Nullfunktion ist. Die dritte Gleichung gibt an, dass  $V_m$  eine skalierte Version des Grundraumes  $V_0$  ist.

Des Weiteren gibt es eine Funktion  $\varphi \in L^2(\mathbb{R})$ , deren ganzzahlige Verschiebungen eine Riesz-Basis von  $V_0$  erzeugen [19].

$$V_0 = \overline{\text{span}\{\varphi(\cdot - k) | k \in \mathbb{Z}\}} \quad (4.8)$$

mit

$$A \sum_{k \in \mathbb{Z}} c_k^2 \leq \left\| \sum_{k \in \mathbb{Z}} c_k \varphi(\cdot - k) \right\|_{L^2}^2 \leq B \sum_{k \in \mathbb{Z}} c_k^2 \quad (4.9)$$

$\forall \{c_k\}_{k \in \mathbb{Z}} \in \ell^2(\mathbb{Z})$  und  $A, B$  positive Konstanten.

Bei der Ungleichung (4.9) handelt es sich um eine Normäquivalenz, somit ist Stabilität gegeben. Gilt  $A = B = 1$ , so bilden die Translate von  $\varphi$  eine orthonormale Basis. Durch einen Orthogonalisierungsprozess kann zu jeder Riesz-Basis mindestens eine Orthonormalbasis gefunden werden [14]. In der hier vorliegenden Arbeit werden ausschließlich orthonormale Basen betrachtet, so dass von nun an  $\varphi$  die durch den Orthogonalisierungsprozess entstandene Funktion bezeichnet, deren ganzzahlige Verschiebungen eine Orthonormalbasis bilden.

Die Gleichung (4.8) impliziert, dass der Raum  $V_0$  invariant bezüglich ganzzahliger Verschiebungen ist (translationsinvariant).

$$\forall k \in \mathbb{Z} \quad f \in V_0 \Leftrightarrow f(\cdot - k) \in V_0 \quad (4.10)$$

Zusammen mit Gleichung (4.7) ergibt sich somit aus (4.10)

$$\forall k \in \mathbb{Z} \quad f(\cdot - 2^m k) \in V_m \Leftrightarrow f \in V_m \quad (4.11)$$

### 4.3.2 Skalierungsfunktion

Aus Gleichung (4.11) können die Funktionen

$$\varphi_{m,k}(t) = 2^{-\frac{m}{2}} \varphi(2^{-m}t - k), \quad k \in \mathbb{Z} \quad (4.12)$$

abgeleitet werden, die eine Orthonormalbasis von dem Raum  $V_m$  sind und diesen aufspannen [6, 7, 8, 19].

$$V_m = \overline{\text{span}\{\varphi_{m,k} | k \in \mathbb{Z}\}} \quad (4.13)$$

Die Funktion  $\varphi$  wird Skalierungsfunktion genannt, da sie die Skalierungsgleichung erfüllt [6, 7, 8, 19], d.h. es gibt eine Folge  $\{h_k\}_{k \in \mathbb{Z}} \in \mathbb{R}$  mit

$$\varphi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2x - k). \quad (4.14)$$

Dies gilt, da nach Gleichung (4.8)  $\varphi \in V_0 \subset V_{-1}$ . Der Raum  $V_{-1}$  wird von den Funktionen  $\varphi_{-1,k}(t) = \sqrt{2}\varphi(2t - k)$  aufgespannt (siehe Gleichungen (4.12) und (4.13)). Jede Funktion aus dem Raum  $V_{-1}$  kann nun als Linearkombination dieser Basisfunktionen dargestellt werden, was Gleichung (4.14) besagt.

Aus Gleichung (4.8) und (4.14) folgt aber nach [10] direkt

$$\varphi_{m,k} = \sum_{l \in \mathbb{Z}} h_l \varphi_{m-1,l+2k} = \sum_{l \in \mathbb{Z}} h_{l-2k} \varphi_{m-1,l}. \quad (4.15)$$

Weiterhin ist  $\varphi$  integrierbar und erfüllt die Mittelungseigenschaft

$$\int_{-\infty}^{\infty} \varphi(t) dt = 1. \quad (4.16)$$

Die Koeffizienten der Skalierungsfunktion genügen den Gleichungen

$$\sum_{k \in \mathbb{Z}} h_k = \sqrt{2} \quad \text{und} \quad \sum_{k \in \mathbb{Z}} (-1)^k h_k = 0. \quad (4.17)$$

### 4.3.3 Wavelet

Es existiert zu jeder Skalierungsfunktion  $\varphi$  innerhalb einer MSA ein Wavelet  $\psi$ , dessen translatierte und dilatierte Versionen

$$\psi_{m,k}(t) = 2^{-\frac{m}{2}} \psi(2^{-m}t - k) \quad (4.18)$$

für ein festes  $m \in \mathbb{Z}$  eine Basis des Raumes  $W_m$  bilden [6, 7, 8, 19, 20]. Darüber hinaus kann das Wavelet mit Hilfe der Skalierungsfunktion und der Folge  $\{g_k\}_{k \in \mathbb{Z}}$  dargestellt werden.

$$\psi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \varphi(2t - k) \quad (4.19)$$

Auch hier ist, analog zu Gleichung (4.15),

$$\psi_{m,k} = \sum_{l \in \mathbb{Z}} g_l \psi_{m-1,l+2k} = \sum_{l \in \mathbb{Z}} g_{l-2k} \psi_{m-1,l} \quad (4.20)$$

resultierend aus den Gleichungen (4.18), (4.19) und der Definition von  $\varphi_{m,k}$  (4.12) [10]. Wie die Skalierungsfunktion soll auch das Wavelet integrierbar sein und die Eigenschaft

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (4.21)$$

besitzen (Vergleiche Gleichung (4.2) auf Seite 34). Um dies zu erfüllen, muss das Wavelet eine oszillierende Funktion sein, also wellenartigen Charakter besitzen: hieraus ist die englische Bezeichnung “Wavelet” entstanden.

Die Folge  $\{g_k\}_{k \in \mathbb{Z}}$  wird mit Hilfe der Koeffizienten der Skalierungsfunktion  $h_k$  durch

$$g_k = (-1)^k h_{1-k}$$

definiert [10, 6, 7, 8, 20]. Durch Integration von  $\psi$  sowie unter Beachtung der Gleichungen mit den Integraleigenschaften (4.16 und 4.21) ist die Bedingung

$$\sum_{k \in \mathbb{Z}} g_k = 0$$

ableitbar.

## 4.4 Schnelle Wavelet-Transformation

### 4.4.1 Transformation

Für die Analyse wird eine orthogonale Skalierungsfunktion  $\varphi$ , die eine Basis von  $V_0$  bildet, gewählt. Somit besitzt  $f \in V_0$  die Darstellung

$$f(x) = \sum_{k \in \mathbb{Z}} c_k^0 \varphi(x - k)$$

mit den reellen Koeffizienten  $c^0 = \{c_k^0 | k \in \mathbb{Z}\}$ .

Das zur Skalierungsfunktion  $\varphi$  gehörige orthogonale Wavelet sei  $\psi$ . Damit ist

$$\{\psi_{m,k} = 2^{-m/2} \psi(2^{-m} \cdot -k) | k, m \in \mathbb{Z}\}$$

eine Orthonormalbasis des  $L^2(\mathbb{R})$  [6, 7, 8, 19, 20].

Die diskrete Wavelet-Transformation von  $f$  erfolgt durch die  $L^2$ -Skalarprodukte von  $f$  und  $\psi_{m,k}$  (dies entspricht der Gleichung (4.3) aus der CWT):

$$\sqrt{c_\psi}(L_\psi f(x))(2^m, 2^m k) = \langle f, \psi_{m,k} \rangle_{L^2} \quad m \in \mathbb{N}_0, k \in \mathbb{Z}$$

.

Zur Vereinfachung der Formulierung eines schnellen, effizienten Algorithmus für die Transformation werden folgende Definitionen eingeführt:

$$d_k^m := \langle f, \psi_{m,k} \rangle_{L^2} \quad , \quad d^m := \{d_k^m | k \in \mathbb{Z}\} \in \ell^2(\mathbb{Z})$$

$$c_k^m := \langle f, \varphi_{m,k} \rangle_{L^2} \quad , \quad c^m := \{c_k^m | k \in \mathbb{Z}\} \in \ell^2(\mathbb{Z})$$

Zusammen mit den Skalierungsgleichungen für die Skalierungsfunktion (4.14) und das Wavelet (4.19) folgt:

$$d_k^m := \langle f, \psi_{m,k} \rangle_{L^2} = \sum_{l \in \mathbb{Z}} g_l \langle f, \varphi_{m-1, 2k-l} \rangle_{L^2} = \sum_{l \in \mathbb{Z}} g_{l-2k} c_l^{m-1} \quad (4.22)$$

$$c_k^m := \langle f, \varphi_{m,k} \rangle_{L^2} = \sum_{l \in \mathbb{Z}} h_l \langle f, \varphi_{m-1, 2k-l} \rangle_{L^2} = \sum_{l \in \mathbb{Z}} h_{l-2k} c_l^{m-1} \quad (4.23)$$

Ist  $c^0$  gegeben, so kann die DWT rekursiv durch die diskrete Faltung berechnet werden. Sämtliche Operationen werden nun auf den Koeffizienten  $c^m$  und  $d^m$  ausgeführt [7, 8, 20]. An dieser Stelle sei angemerkt, dass es sich nicht um eine normale diskrete Faltung handelt, da zum einen nur jedes zweite Skalarprodukt ausgewertet wird und zum anderen die Indizes vertauscht sind ( $l-2k$  statt  $k-2l$ ) [6, 8, 19]. In der Signaltheorie wird von einer Faltung mit anschließendem Sub-Sampling um den Faktor 2 gesprochen.

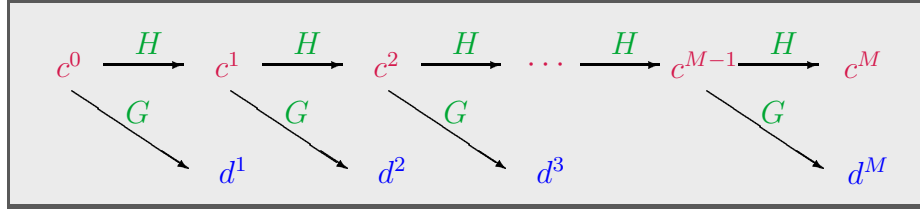
Graphisch ist das Prinzip der schnellen Wavelet Transformation in Abbildung 4.4 dargestellt. Hierbei wurden zur kürzeren Schreibweise der Gleichungen (4.22) und (4.23) die Zerlegungsoperatoren  $G$  und  $H$  mit

$$\begin{aligned} G &: \ell^2(\mathbb{Z}) \rightarrow \ell^2(\mathbb{Z}) \\ c &\mapsto Gc = \left\{ (Gc)_k = \sum_{l \in \mathbb{Z}} g_{l-2k} c_l \right\} \end{aligned}$$

bzw.

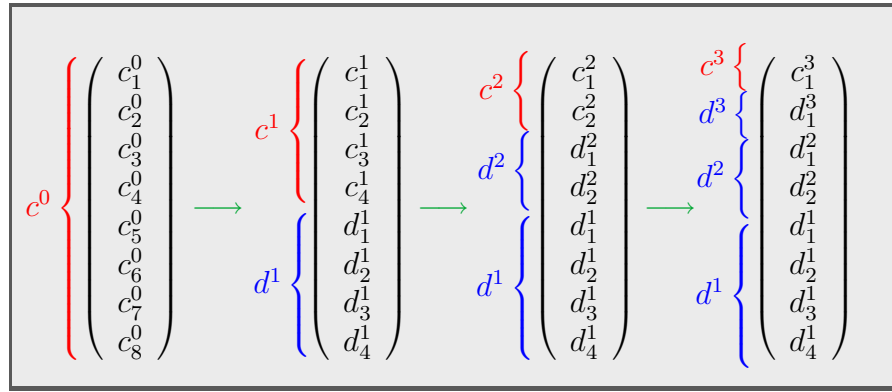
$$\begin{aligned} H &: \ell^2(\mathbb{Z}) \rightarrow \ell^2(\mathbb{Z}) \\ c &\mapsto Hc = \left\{ (Hc)_k = \sum_{l \in \mathbb{Z}} h_{l-2k} c_l \right\} \end{aligned}$$

eingeführt, wobei  $h = \{h_k | k \in \mathbb{Z}\}$  bzw.  $g = \{g_k | k \in \mathbb{Z}\}$  die Folge der Skalierungskoeffizienten bzw. die Folge der Wavelet-Koeffizienten bezeichnen.



**Abbildung 4.4:** Schematische Darstellung einer schnellen Wavelet-Transformation: Die diskrete Faltung mit der Skalierungsfunktion bzw. dem Wavelet mit anschließendem Sub-Sampling um Faktor 2 wird durch die beiden Operatoren  $H$  bzw.  $G$  dargestellt.

Dieser Algorithmus ist als Mallat-Algorithmus oder auch als Burt-Adelsonscher Pyramiden-Algorithmus bekannt. Anschaulicher wird er und seine Pyramidenstruktur, wenn er auf einen Datenvektor angewendet wird [8, 16, 19, 20].



**Abbildung 4.5:** Pyramiden-Algorithmus angewendet auf einen Datenvektor mit 8 Elementen und einer maximalen Zerlegungstiefe von  $M = 3$ .

#### 4.4.2 Rekonstruktion

Bei der Rekonstruktion wird aus den Koeffizientenfolgen  $\{c^m, d^m | m = 1, \dots, M\}$  die Folge  $c^0$  berechnet. Durch die Pyramidenstruktur der Transformation ist es offensichtlich, dass es sich um ein rekursives Problem handelt. Es reicht zu zeigen, wie aus  $c^1$  und  $d^1$  die Folge  $c^0$  ermittelt wird.

Da  $c^0 \in V_0 = V_1 \oplus W_1$ , gilt zusammen mit den Gleichungen (4.15) und (4.20), resultierend aus den Skalierungsgleichungen (4.14) und (4.19), folgendes:

$$\begin{aligned} \sum_{k \in \mathbb{Z}} c_k^0 \varphi_{0,k} &= \sum_{j \in \mathbb{Z}} c_j^1 \varphi_{1,j} + \sum_{j \in \mathbb{Z}} d_j^1 \psi_{1,j} \\ &= \sum_{j \in \mathbb{Z}} c_j^1 \sum_{l \in \mathbb{Z}} h_l \varphi_{0,2j-l} + \sum_{j \in \mathbb{Z}} d_j^1 \sum_{l \in \mathbb{Z}} g_l \varphi_{0,2j-l} \end{aligned}$$



Mit Hilfe eines Koeffizientenvergleichs ergibt sich  $c^0$  zu:

$$c_k^0 = \sum_{l \in \mathbb{Z}} c_l^1 h_{2l-k} + \sum_{l \in \mathbb{Z}} d_l^1 g_{2l-k}$$

Ausgehend von  $d^M$  und  $c^M$  kann nun analog zunächst  $c^{M-1}$  rekonstruiert werden. Rekursiv angewendet gelangt führt dies nach  $M$  Schritten zu  $c^0$  [19].

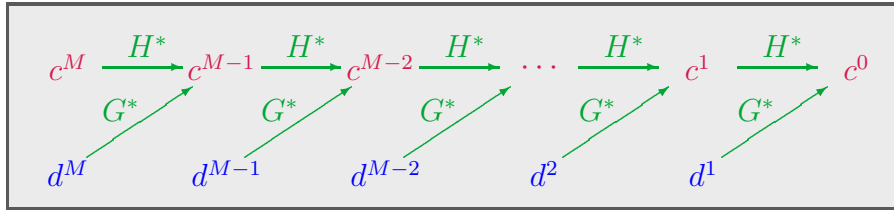
Analog können an dieser Stelle die zu den Operatoren  $G$  und  $H$  bezüglich des  $L^2$ -Skalarproduktes adjungierten Operatoren  $G^*$  und  $H^*$  definiert werden.

$$\begin{aligned} G^* &: \ell^2(\mathbb{Z}) \rightarrow \ell^2(\mathbb{Z}) \\ c &\mapsto G^*c = \left\{ (G^*c)_k = \sum_{l \in \mathbb{Z}} g_{k-2l} c_l \right\} \end{aligned}$$

bzw.

$$\begin{aligned} H^* &: \ell^2(\mathbb{Z}) \rightarrow \ell^2(\mathbb{Z}) \\ c &\mapsto H^*c = \left\{ (H^*c)_k = \sum_{l \in \mathbb{Z}} h_{k-2l} c_l \right\} \end{aligned}$$

Somit kann die schnelle Rekonstruktion analog zur schnellen Transformation graphisch wie folgt dargestellt werden.



**Abbildung 4.6:** Schematische Darstellung einer schnellen Rekonstruktion: Die Operatoren  $H^*$  bzw.  $G^*$  sind hierbei die adjungierten Operatoren zu  $H$  bzw.  $G$  bezüglich des  $L^2$ -Skalarproduktes.

### 4.4.3 Komplexität

Bevor an einem konkreten Beispiel die schnelle Wavelet-Transformation veranschaulicht wird, wird zunächst die Komplexität dieses Verfahrens betrachtet [19].

Die Länge von  $c^0$  wird mit  $n(0) = n$  und die Anzahl der Koeffizienten der Skalierungsfunktion bzw. des Wavelets mit  $g$  bzw.  $h$  bezeichnet. Da in jedem Schritt

nach der Faltung ein Sub-Sampling um Faktor 2 vorgenommen wird, berechnet sich die Länge von  $c^m$  bzw.  $d^m$  durch  $n(m) = 2^{-m} \cdot n(0)$ .

Zugleich finden in jedem Schritt zwei Faltungen statt, einmal mit der Skalierungsfunktion und einmal mit dem Wavelet. Unter der Annahme, dass  $g$ ,  $h$  und  $n$  endlich sind, beläuft sich somit der Gesamtaufwand dieses Verfahrens bei einer Zerlegungstiefe  $M$  auf

$$\sum_{m=1}^M ((g + h) \cdot n(m)) = (g + h) \cdot n(0) \cdot \sum_{m=1}^M 2^{-m} \leq (g + h) \cdot n.$$

Die Komplexität ist proportional zu  $n$ , der Algorithmus weist ein  $O(n)$ -Verhalten auf.

#### 4.4.4 Beispiel: Haar-Wavelet

Das Haar-Wavelet  $\psi_H$  bildet eine orthogonale Wavelet-Basis.

$$\psi_H(x) := \begin{cases} 1 & : 0 \leq x < 0.5 \\ -1 & : 0.5 \leq x \leq 1 \\ 0 & : \text{sonst} \end{cases}$$

$$\varphi_H(x) := \begin{cases} 1 & : 0 \leq x \leq 1 \\ 0 & : \text{sonst} \end{cases}$$

Bedingt durch seine nichtstetige Definition hat das Haar-Wavelet allerdings schlechte Glättungseigenschaften. Auch ist es im Frequenzraum schlecht lokalisiert. Obwohl es diese Nachteile hat, eignet es sich auf Grund seiner Einfachheit sehr gut, um eine Wavelet-Transformation und -Rekonstruktion zu demonstrieren.

Die Koeffizienten der Skalierungsfunktion  $\varphi_H$  sind durch  $h_k = 1/\sqrt{2}$ ,  $k \in \{1, 2\}$ , die des Wavelets  $\psi_H$  durch  $g_k = (-1)^{k+1} \cdot 1/\sqrt{2}$ ,  $k \in \{1, 2\}$ , gegeben. Somit glättet die Skalierungsfunktion  $\varphi_H$  das Signal, indem es die Mittelwerte aus je zwei benachbarten Werten bildet und mit  $\sqrt{2}$  multipliziert. Man erhält ein Signal mit halber Auflösung. Das Wavelet  $\psi_H$  berechnet die Differenzen zum Mittelwert multipliziert mit  $\sqrt{2}$  [6, 7, 8]. An dieser Stelle sei angemerkt, dass die Multiplikation mit dem Faktor  $\sqrt{2}$  zur Erhaltung der Signalenergie notwendig ist.

Es seien die Werte  $c^0 = [2, 2, 5, 3]$  gegeben. Die Skalierungsfunktion berechnet den Mittelwert aus den ersten und den letzten beiden Werten, das Wavelet die entsprechenden Differenzen:

$$\begin{aligned} \left[ \frac{2+2}{2} \cdot \sqrt{2}, \frac{5+3}{2} \cdot \sqrt{2} \right] &= [2\sqrt{2}, 4\sqrt{2}] \\ \left[ \frac{2-2}{2} \cdot \sqrt{2}, \frac{5-3}{2} \cdot \sqrt{2} \right] &= [0, \sqrt{2}] \end{aligned}$$

Auf den ersten Vektor  $[2\sqrt{2}, 4\sqrt{2}]$  kann die Transformation erneut angewendet werden:

$$\begin{aligned} \left[ \frac{2\sqrt{2} + 4\sqrt{2}}{2} \cdot \sqrt{2} \right] &= [6] \\ \left[ \frac{2\sqrt{2} - 4\sqrt{2}}{2} \cdot \sqrt{2} \right] &= [-2] \end{aligned}$$

Die nach der Transformation abgespeicherten Werte sind also  $[6, -2, 0, \sqrt{2}]$ .

Zur Rekonstruktion wird zunächst die letzte Transformation aufgelöst, indem der Mittelwert bzw. die Differenz zum Mittelwert der beiden zuletzt errechneten Werte berechnet und mit  $\sqrt{2}$  multipliziert wird.

$$\left[ \frac{6+2}{2} \cdot \sqrt{2}, \frac{6-2}{2} \cdot \sqrt{2} \right] = [2\sqrt{2}, 4\sqrt{2}]$$

Nun werden die ursprünglichen Daten durch eine weitere Rekonstruktion ermittelt. Man geht genauso vor, wie bei der Auflösung der vorherigen Synthese: Aus beiden Vektoren ( $[2\sqrt{2}, 4\sqrt{2}]$  und  $[0, \sqrt{2}]$ ) wird jeweils der erste Wert herausgenommen und zu diesen der Mittelwert bzw. die Differenz zum Mittelwert ermittelt und jeweils noch mit  $\sqrt{2}$  multipliziert. Ebenso wird mit jedem weiteren folgenden Wertepaar verfahren:

$$\left[ \frac{2\sqrt{2}+0}{2} \cdot \sqrt{2}, \frac{2\sqrt{2}-0}{2} \cdot \sqrt{2}, \frac{4\sqrt{2}+\sqrt{2}}{2} \cdot \sqrt{2}, \frac{4\sqrt{2}-\sqrt{2}}{2} \cdot \sqrt{2} \right] = [2, 2, 5, 3]$$

## 4.5 Eigenschaften von Wavelets

Die verschiedenen Basis-Wavelets zeichnen sich durch unterschiedliche Kombinationen von Eigenschaften aus. Basis-Wavelets mit gleichen Eigenschaften werden zu Klassen zusammengefasst [18]. Die wichtigsten Eigenschaften sind:

1. *Orthogonalität:* In dieser Arbeit wurden nur orthogonale Wavelets verwendet.
2. *Kompaktheit:* Waveletfunktionen mit kurzem Träger (Support) gewährleisten gute Lokalitätseigenschaften. Ein kompaktes Wavelet, ein Wavelet mit kurzem Träger, hat nur wenige Koeffizienten  $g_k$  ungleich Null, wodurch der Rechenaufwand bei der Verwendung eines solchen Wavelets stark begrenzt ist.

3. *Glattheit*: Die Glattheit ist ein notwendiges Kriterium bei Approximations- und Kompressionsverfahren. Ein Kompressionseffekt wird erzielt, indem Koeffizienten unterhalb eines Schwellenwertes zu Null gesetzt werden und somit die dazugehörigen Basisfunktionen nicht mehr zur Rekonstruktion verwendet werden (siehe hierzu auch Kapitel 4.7 auf Seite 50). Mit der Verwendung glättender Wavelets kann die Entstehung störender Artefakte in rekonstruierten Signalen vermindert werden.
4. *Symmetrie*: Wavelets können zur y-Achse symmetrisch sein - wie z. B. der Mexikanerhut (Abbildung 4.1 auf Seite 35) - oder asymmetrisch, wie z.B. die Daubechies Wavelets (siehe Abbildung 4.2 ebenfalls auf Seite 35). Asymmetrische Basisfunktionen haben die Eigenschaft, dass sie zu einer Phasenverschiebung des Signals führen können.
5. *Verschwindende Momente*: Das  $k$ -te Moment einer Funktion ist das Integral über das Produkt dieser Funktion und der  $k$ -ten Potenz der Integrationsvariablen.

$$m_k = \int t^k \psi(t) dt \quad k = 0, 1, \dots$$

Ist  $m_k = 0$ , so wird von einem verschwindenden Moment gesprochen. Die Ordnung  $N$  eines Daubechies Wavelets gibt an, welches Moment das erste nicht verschwindende Moment ist [6, 7, 8, 19]:

$$\begin{aligned} m_k &= \int t^k \psi(t) dt = 0 & k = 0, 1, \dots, N-1 \\ m_N &= \int t^N \psi(t) dt \neq 0 \end{aligned} \quad (4.24)$$

Im diskreten Fall wird das Integral durch eine Summe ersetzt.

Neben diesen Eigenschaften gibt es noch eine Reihe anderer Unterscheidungskriterien, die jedoch unwesentliche Bedeutung für diese Arbeit haben. Da sich diese Eigenschaften oftmals gegenseitig ausschließen, ist die Suche nach der optimalen Kombination solcher Eigenschaften für eine gegebene Anwendung recht schwierig.

Eine besondere Wavelet-Familie sind die orthonormalen Wavelet-Basen. Diese Art von Wavelets bietet den Vorteil, dass die Umkehrabbildung der Transformation, die Rekonstruktion, leicht zu ermitteln ist. Stellt man die Transformation mit Hilfe einer großen Matrix dar, so ist die Umkehrabbildung - die Inverse also - gleich der Transponierten. Das einfachste Wavelet dieser Familie ist das Haar-Wavelet  $\psi_H$ , welches allerdings aufgrund seiner unstetigen Definition schlechte Glättungseigenschaften besitzt. Das wohl komplizierteste Wavelet dieser Klasse ist das Meyer-Wavelet, welches zwar hervorragende Glättungseigenschaften, allerdings auch einen unendlichen Support besitzt, so dass es für reale Anwendungen ungeeignet ist.

Interessanter sind daher die von Ingrid Daubechies entdeckten kompakten, differenzierbaren, orthogonalen Wavelets. Sie besitzen nicht nur eine hohe Lokalität im Frequenz- und Zeitraum, sondern haben auch noch einen endlichen Support. Zusätzlich können sie so glatt gewählt werden wie es gerade nötig ist. Das Daubechies-Wavelet *Daub2*, welches dem Haar-Wavelet entspricht, bildet lineare Funktionen exakt ab. Es hat ein verschwindendes Moment.

## 4.6 Daubechies Wavelet Filterkoeffizienten

Den Daubechies Wavelet Filterkoeffizienten liegt eine spezielle Klasse von Wavelets zugrunde. Die Bezeichnung “Filter” stammt aus der digitalen Signalverarbeitung: Ein Wavelet entspricht der Impulsantwort eines Bandpassfilters, das aus einem Hoch- und einem Tiefpass zusammengesetzt ist, mit einer gewissen Schärfe in der Zeit (Filterlänge) und in der Frequenz (Bandbreite) [10]. Wie man bei der Darstellung mit Filtern bei der diskreten Wavelet-Transformation vorgeht, soll nun im Folgenden erklärt werden:

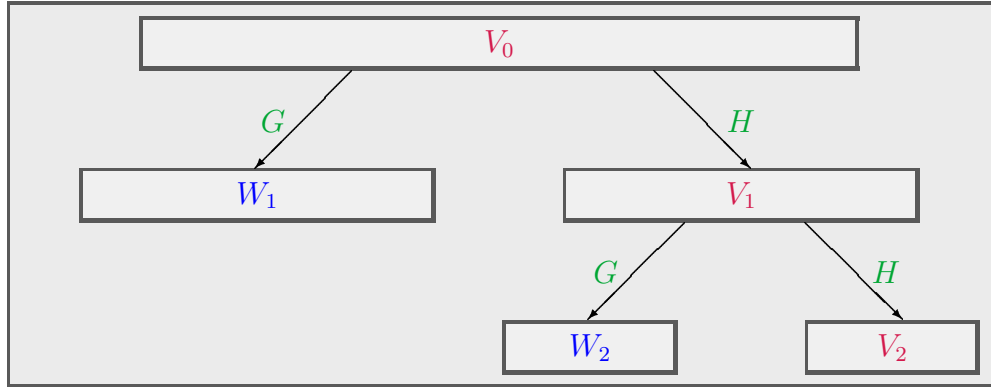
Zuerst wird ein Tiefpassfilter, das mit der Skalierungsfunktion verbunden ist, über das Signal geschoben. Dabei verändert sich die Position des Filters jeweils um den Faktor zwei. An jeder Position wird dieser mit den darunter liegenden Signalwerten gefaltet, d.h. multipliziert und dann aufaddiert. Die mathematische Formel für die Faltung ist

$$(h * c)_k = \sum_{j=0}^k h_j c_{k-j} \quad \text{mit } h_j, c_j > 0 \quad \forall j$$

wobei  $h$  die Koeffizienten des Filters,  $c$  die Werte des Signals und  $k$  die Anzahl der Signalwerte darstellen. Dies entspricht genau den Gleichungen (4.22) bzw. (4.23) auf Seite 41. Das Tiefpassfilter hat die Eigenschaft, hohe Frequenzen aus dem Signal herauszufiltern und nur die niedrigen Frequenzen “passieren” zu lassen. So erhält man ein geglättetes Signal, das dem Ausgangssignal in einer größeren Auflösung entspricht und nur halb so viele Werte wie dieses besitzt [10].

Auf dieselbe Weise wird ein Hochpassfilter über das Signal gelegt. Dieses Filter entspricht stark gestauchten Wavelets. Es läßt nur die hohen Frequenzen “passieren” und filtert die niederen heraus. Um das Ursprungssignal zu rekonstruieren, müssten diese Werte dem geglätteten Signal hinzugefügt werden [10].

Im nächsten Schritt wiederholt man das gesamte Verfahren mit dem geglätteten Signal. Dieses wird also wieder in zwei Teile zerlegt, in den Tiefpass- und den Hochpass-Teil. Man fährt rekursiv fort. Die Detailwerte jeder Transformationsstufe werden dabei festgehalten.



**Abbildung 4.7:** Schematische Darstellung der diskreten Wavelet-Transformation mit Hilfe von Filtern: Der Hochpass wird mit  $H$ , der Tiefpass mit  $G$  bezeichnet. Diese Abbildung entspricht im Wesentlichen der Abbildung 4.4 auf Seite 42.

Um die Filterkoeffizienten bzw. an die Koeffizienten der Skalierungsfunktion und des Wavelets zu ermitteln, müssen lediglich die Skalierungsgleichungen gelöst werden. Im Falle der Daubechies Koeffizienten kann dies mit Hilfe von Matrizen erfolgen [13]. Zur Erleichterung der Notation wird dies anhand eines einfachen und sehr lokalisierten Vertreters dieser Klasse, welcher oftmals mit  $Daub_4$ <sup>1</sup> bezeichnet wird, aufgezeigt. Die Transformationsmatrix für diesen Fall sieht folgendermaßen aus:

$$W = \begin{pmatrix} c_0 & c_1 & c_2 & c_3 & & & & \\ c_3 & -c_2 & c_1 & -c_0 & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & \\ & & c_3 & -c_2 & c_1 & -c_0 & & \\ & & & & \ddots & & & \\ & & & & & c_0 & c_1 & c_2 & c_3 \\ & & & & & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & & & & & & c_0 & c_1 \\ c_1 & -c_0 & & & & & & c_3 & -c_2 \end{pmatrix}$$

Der Aufbau der Matrix ist recht simpel. In den ungeraden Zeilen stehen die Filterkoeffizienten in aufsteigender Reihenfolge, beginnend bei der Hauptdiagonalen. In den geraden Zeilen stehen die Koeffizienten in absteigender Reihenfolge, wobei jeder zweite Koeffizient zusätzlich mit -1 multipliziert wird, beginnend in der gleichen Spalte wie in der vorherigen Zeile. Alle noch freien Elemente sind gleich Null. Eine Besonderheit bilden die letzten beiden Zeilen, deren letzte beiden Koeffizienten wie bei periodischen Randbedingungen den rechten Matrixrand

<sup>1</sup>Eine Abbildung dieses Wavelets und seiner Skalierungsfunktion befindet sich im Anhang A

überschreiten und vom linken Rand wieder in der Matrix erscheinen.

$$W^T = \begin{pmatrix} c_0 & c_3 & & & & & c_2 & c_1 \\ c_1 & -c_2 & & & & & c_3 & -c_0 \\ c_2 & c_1 & c_0 & c_3 & & & & \\ c_3 & -c_0 & c_1 & -c_2 & & & & \\ & & & \ddots & & & & \\ & & & & c_2 & c_1 & c_0 & c_3 \\ & & & & c_3 & -c_0 & c_1 & -c_2 \\ & & & & & & c_2 & c_1 & c_0 & c_3 \\ & & & & & & c_3 & -c_0 & c_1 & -c_2 \end{pmatrix}$$

Die hier betrachtete Klasse von Wavelets hat die Eigenschaft, orthogonal zu sein. Daher muss die Transponierte  $W^T$  gleich der Inversen  $W^{-1}$  sein. Dies führt zu den Bedingungen:

$$\begin{aligned} c_0^2 + c_1^2 + c_2^2 + c_3^2 &= 1 \\ c_2 c_0 + c_3 c_1 &= 0 \end{aligned}$$

Da die Ordnung  $p = 2$  erreicht wird, kommen zu den beiden obigen Gleichungen noch die folgenden zwei hinzu, herrührend von den Gleichungen (4.17) und (4.24):

$$\begin{aligned} c_3 - c_2 + c_1 - c_0 &= 0 \\ 0c_3 - 1c_2 + 2c_1 - 3c_0 &= 0 \end{aligned}$$

Es entsteht ein lineares Gleichungssystem mit vier Gleichungen und vier Unbekannten, welche eindeutig bestimmt werden können. Die Lösung lautet:

$$\begin{aligned} c_0 &= (1 + \sqrt{3})/4\sqrt{2} \\ c_1 &= (3 + \sqrt{3})/4\sqrt{2} \\ c_2 &= (3 - \sqrt{3})/4\sqrt{2} \\ c_3 &= (1 - \sqrt{3})/4\sqrt{2} \end{aligned}$$

Für ein Daubechies Wavelet höherer Ordnung müssen die entsprechenden Koeffizienten bestimmt werden, indem die obige Matrix nach beschriebenem Muster aufgebaut wird, die zu erfüllenden Momentbedingungen (4.24) aufgestellt werden und das so entstehende Gleichungssystem gelöst wird. Zu beachten ist hierbei, dass die Anzahl der Koeffizienten um zwei steigt, wenn die Ordnung  $p$  um eins erhöht wird. Bis zu einer Ordnung von 10 sind die Koeffizienten numerisch tabelliert. Die Koeffizienten der in dieser Arbeit untersuchten Wavelets sind im Anhang A zu finden.

## 4.7 Wavelet-Thresholding

Die Wavelet-Transformation wird gerne zur Datenkompression eingesetzt. Eine Kompression wird erreicht, indem die Werte der transformierten Daten, die unterhalb eines Schwellenwertes  $t$  liegen, gleich Null gesetzt werden (Thresholding) [20]. Dies führt zu einer Veränderung der Originaldaten und damit unweigerlich zu einem Fehler. Wie groß dieser Fehler ist, hängt von der Höhe des Schwellenwertes ab. Im Wesentlichen gibt es zwei verschiedene Threshold-Verfahren: Das Hard- und das Soft-Thresholding.

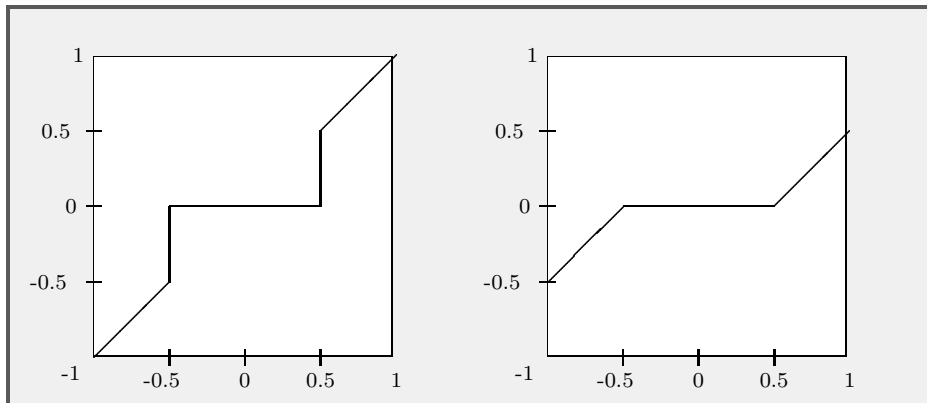
Bei dem Hard-Thresholding werden alle transformierten Daten, die oberhalb des Schwellenwertes  $t$  liegen, beibehalten, während alle anderen auf Null gesetzt werden. Es wird nach dem Prinzip “behalten oder abschneiden” vorgegangen [20].

$$\eta_t^{\text{hard}}(v) = I(|v| > t) \cdot v = \begin{cases} v & : |v| > t \\ 0 & : |v| \leq t \end{cases}$$

Wird davon ausgegangen, dass die Eingangsdaten bereits fehlerhaft sind (in der Signaltheorie spricht man von Rauschen), ist es sinnvoll, alle Koeffizienten leicht zu verändern. Diese Überlegung führt zum Soft-Thresholding.

$$\eta_t^{\text{soft}}(v) = I(|v| > t) \cdot (|v| - t) \cdot \text{sgn}(v) = \begin{cases} \text{sgn}(v) \cdot (|v| - t) & : |v| > t \\ 0 & : |v| \leq t \end{cases}$$

Hierbei werden wie zuvor alle transformierten Daten unterhalb eines Schwellenwertes  $t$  auf Null gesetzt, alle anderen Daten werden um den Schwellenwert verringert bzw. bei negativen Werten erhöht. Das Motto lautet hier “verkleinern oder abschneiden” [20]. Ein Beispiel für beide Thresholdingarten ist in Abbildung 4.8 dargestellt.



**Abbildung 4.8:** Thresholding bei der Identität mit Schwellenwert  $t = 0.5$ , links das Hard- und rechts das Soft-Thresholding.



Die Wavelet-Transformation wird rekursiv auf den Daten ausgeführt, womit verschiedene Transformationsebenen entstehen. Man spricht hier auch von Mehrfachauflösung. Zwangsläufig stellt sich die Frage, wann das Treshold-Verfahren angewendet werden soll: Man kann nach jeder Transformationsebene das Treshold-Verfahren anwenden, oder erst nachdem die endgültige Transformationstiefe erreicht ist [20]. Im ersten Fall kann man sich zusätzlich noch zwischen der Möglichkeit entscheiden, ob man einen Schwellenwert für alle Transformationsebenen (globales Thresholding), oder für jede Ebene einen eigenen (levelabhängiges Thresholding) benutzt. Beim globalen Thresholding wendet man meist den Operator nur auf die Koeffizienten der höchsten Auflösung an, da die Koeffizienten der unteren Levels die “größten” sind.

Bei dem Verfahren, das dieser Arbeit zu Grunde liegt, wurde ein globales Thresholding gewählt, welches lediglich auf die höchste Auflösung angewendet wurde, d.h. erst nach dem Erreichen der endgültigen Transformationstiefe. Das Thresholding wird durch den Operator  $T : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^n$  dargestellt.



# Kapitel 5

## Ergebnisse

Ein Vielteilchensystem wird im Wesentlichen durch die Teilchenanzahl  $N$  und die Dichte  $\rho$  charakterisiert. Zusätzliche Charakteristika - wie Position der Teilchen und deren Ladungen - beschreiben lediglich eine mögliche Realisierung eines solchen Systems. Um qualitative Aussagen bezüglich des verwendeten Verfahrens treffen zu können, werden für jede Kombination von Verfahrensparametern mehrere unterschiedliche, zufällige Startkonfigurationen gewählt und aus den Ergebnissen der Mittelwert bestimmt. Gerechnet wurde auf dem Jülicher Supercomputer JuMP<sup>1</sup>.

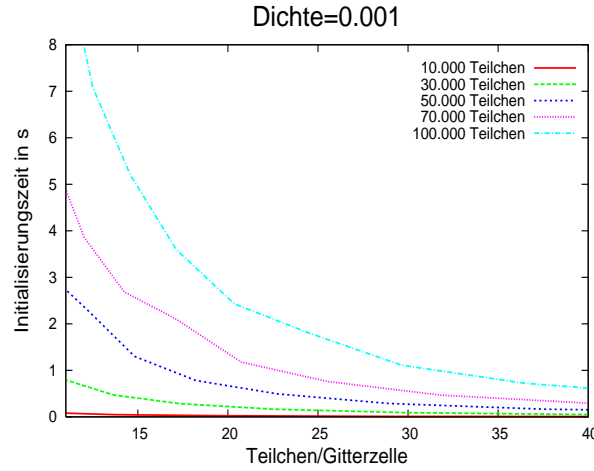
### 5.1 Abschätzung des Diskretisierungsfehlers

Die Genauigkeit des Verfahrens unter Auslassung des Kompressionsverfahrens kann durch vier Parameter gesteuert werden. Die Güte dieser Methode hängt von der benutzten Interpolation, des verwendeten finiten Differenzen-Schemas, der durchschnittlichen Anzahl von Teilchen pro Gitterzelle und der Anzahl der Nachbarzellen für die Nahfeldberechnung ab. Da die Teilchenanzahl im System fest vorgegeben ist, wird zusammen mit der durchschnittlichen Anzahl von Teilchen pro Gitterzelle bestimmt, in wieviele Gitterzellen das Simulationsgebiet unterteilt wird. Viele Teilchen je Gitterzelle bedeuten wenige Gitterzellen, wenige Teilchen pro Zelle dementsprechend viele Gitterzellen.

Da das Laufzeitverhalten im Wesentlichen nur von der Anzahl der Gitterzellen abhängt und die übrigen Parameter kaum einen Einfluss auf den Aufwand ausüben, wird zuerst der Zeitbedarf betrachtet. Bei dem hier verwendeten Verfahren gibt es zum einen den einmaligen Aufwand zu Beginn der Simulation, in der die Abstandsmatrix und die Nachbarschaftslisten aufgebaut werden, und den Zeitbedarf pro Zeitschritt. Je mehr Gitterzellen verwendet werden, desto höher ist der Initialisierungsaufwand. Abbildung 5.1 stellt dies dar.

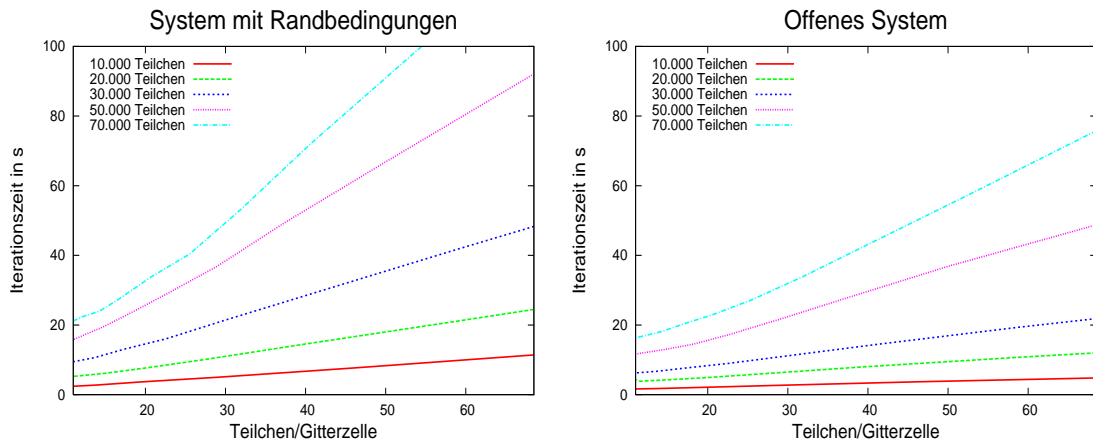
---

<sup>1</sup>Technische Details unter <http://jumpdoc.fz-juelich.de/>



**Abbildung 5.1:** Der Initialisierungsaufwand des Verfahrens ohne das Einfügen der Wavelet-Transformation.

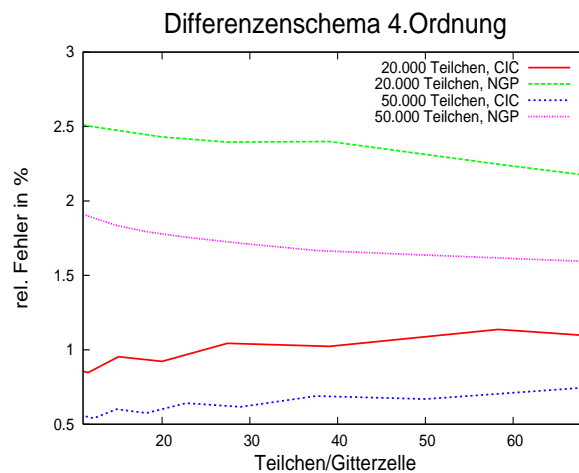
Da dieser Aufwand jedoch nur einmal aufgewendet werden muss, ist er nicht so relevant wie der Zeitbedarf pro Zeitschritt. Wird das System in nur wenige Gitterzellen diskretisiert, so steigt der Aufwand für einen Zeitschritt drastisch an. Dies liegt darin begründet, dass in diesem Fall ein recht hoher Anteil der Wechselwirkungen direkt durch die zeitaufwendige PP-Methode berechnet wird, einem  $O(N^2)$ -Algorithmus. Werden periodische Randbedingungen verwendet, so ist der Aufwand viel größer, als wenn ein offenes System gewählt wird. Die Ursache hierfür ist, dass sich bei offenen Systemen weniger Teilchen im Nahfeld von Randzellen befinden als bei Systemen mit periodischen Randbedingungen und dadurch die zeitaufwendige Berechnung des Nahfeldes in diesem Fall geringer ausfällt.



**Abbildung 5.2:** Der Aufwand des Verfahrens pro Zeitschritt ohne das Einfügen der Wavelet-Transformation, links für ein System mit periodischen Randbedingungen und rechts für ein offenes System.

Wird das Fehlerverhalten des Verfahrens für die verschiedenen Parametereinstellungen untersucht, so verhält sich der Fehler in den Kräften bei Systemen mit periodischen Randbedingungen einerseits und bei offenen Systemen andererseits stark unterschiedlich. Daher werden zunächst die Systeme mit periodischen Randbedingungen betrachtet, anschließend dann die offenen Systeme.

Der verursachte Fehler in den Kräften ist beim NGP-Schema erwartungsgemäß größer als beim CIC-Schema. Auffallend ist, dass der Fehler mit feineren Diskretisierungen beim NGP-Schema noch zunimmt, bei der CIC-Interpolation jedoch abnimmt. Da das CIC-Schema bei gleichem Aufwand bessere Ergebnisse liefert, sollte es dem NGP-Schema vorgezogen werden. Siehe hierzu auch Abbildung 5.3



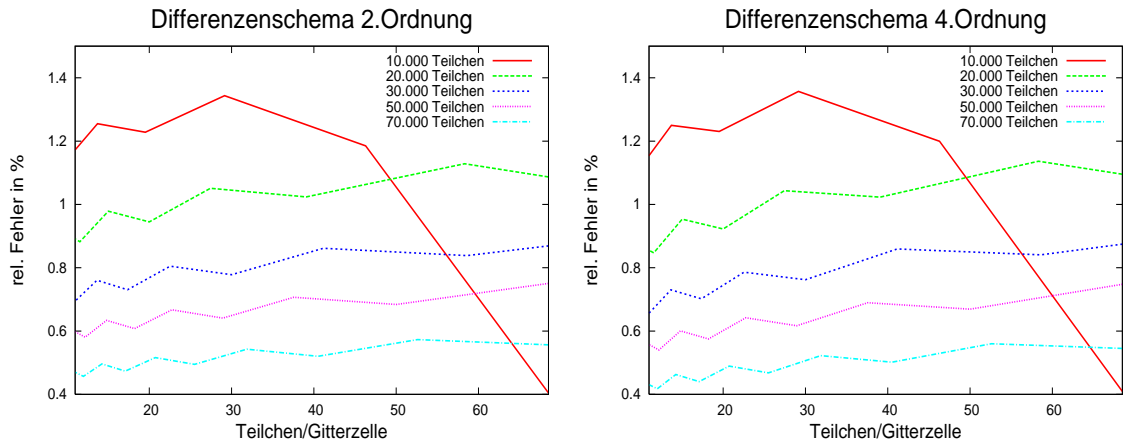
**Abbildung 5.3:** Unterschiedliches Verhalten des Fehlers in den Kräften bei der NGP- und der CIC-Interpolation. Dargestellt sind Systeme mit 20.000 bzw. 50.000 Teilchen, bei denen periodische Randbedingungen benutzt werden.

Denkbar sind natürlich auch andere Interpolationen wie z.B. die Spline-Interpolation. Es ist zu erwarten, dass bei einer Interpolation höherer Ordnung zwar einerseits die Genauigkeit weiter zunimmt, aber andererseits der Aufwand stark ansteigt. Zusätzlich zum Zeitbedarf der Durchführung der Interpolation selber wird durch die Interpolation auch der Aufwand bei der Nahfeldkorrektur bestimmt. Je mehr Gitterpunkte bei der Interpolation der Ladung eines Teilchens einen Ladungsanteil bekommen, desto aufwendiger und komplizierter ist die Nahfeldkorrektur. Das CIC-Schema liefert akzeptable Ergebnisse bei geringem Aufwand.

Wird das System in so viele Gitterzellen unterteilt, dass alle Zellen Nachbarzellen voneinander sind, so ist der Fehler in den Kräften gleich Null, da alle Wechselwirkungen exakt mittels der PP-Methode eingehen. Wird das System mit der nächst feineren Untergliederung diskretisiert, so steigt der Fehler sprunghaft

an, konvergiert aber mit feiner werdender Diskretisierung langsam gegen Null. Auffallend ist, dass bei einer Untergliederung in eine gerade Anzahl von Gitterzellen je Dimensionsrichtung der Fehler geringer ist als bei der nächst feineren Unterteilung, einer ungeraden Anzahl von Zellen je Dimensionsrichtung. Dieser Sachverhalt wird in Abbildung 5.4 gezeigt.

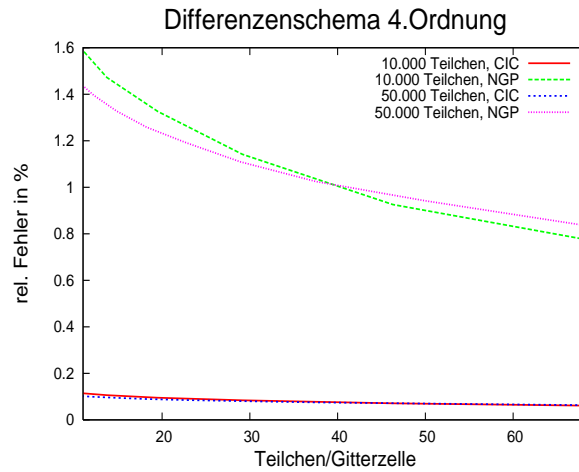
Aus Abbildung 5.4 ist ebenfalls erkennbar, dass bei der Verwendung eines finiten Differenzenschemas höherer Ordnung zur Berechnung des globalen elektrischen Feldes auf den Gitterpunkten, ein geringerer Fehler verursacht wird. Implementiert wurden finite Differenzenschemata 2. Ordnung und 4. Ordnung.



**Abbildung 5.4:** Darstellung des relativen Fehlers in den Kräften, der durch die Diskretisierung verursacht wird, bei einer Dichte von 0.001, der CIC-Interpolation und 125 Nachbarzellen für unterschiedlich viele Teilchen in der Simulation.

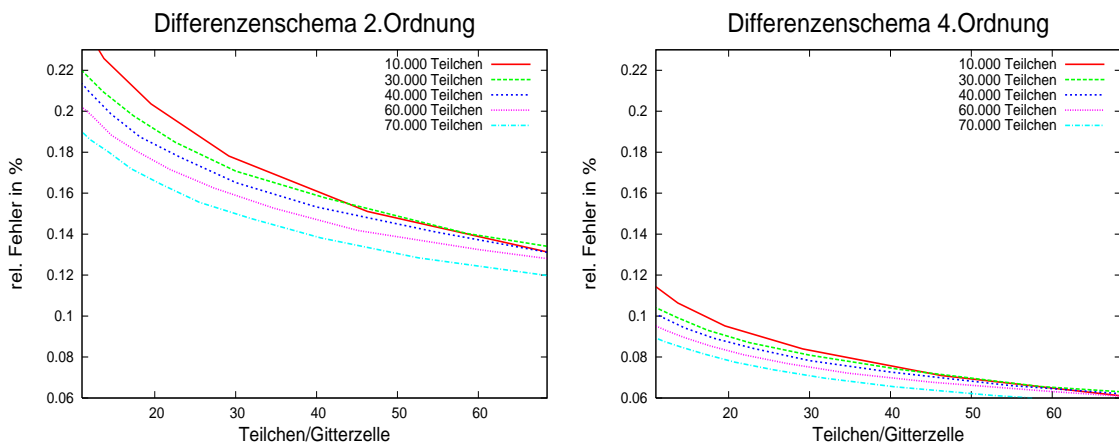
Werden dieselben Untersuchungen für offene Systeme durchgeführt, zeigt sich auch hier, dass das CIC-Schema das geeignetere ist. Der Unterschied in der Höhe des Fehlers bei Verwendung des NGP-Schemas im Vergleich zum CIC-Schema ist im Falle der offenen Systeme sogar viel gravierender. Auch hier steigt der Fehler in den Kräften beim NGP-Schema mit feinerer Diskretisierung an. Dieser Anstieg ist bei offenen Systemen sehr viel stärker ausgeprägt als bei Systemen mit periodischen Randbedingungen.

Diese stärkere Ausprägung des Fehleranstieges bei Verwendung des NGP-Schemas bei feineren Gittern für offene Systeme wird verständlich, wenn das Fehlerverhalten in Abhängigkeit der Gitterunterteilung betrachtet wird. Selbst beim CIC-Schema nimmt der Fehler in den Kräften bei offenen Systemen mit feiner werdender Diskretisierung zu. Der theoretische Grenzfall liegt bei einer extrem feinen Unterteilung des Simulationsgebietes vor. Ist die Untergliederung so fein gewählt, dass es aufgrund des hinreichend kleinen Abstandes zwischen den Teilchen und den Gitterpunkten keine Rolle mehr spielt, ob die Ladungen der Teilchen



**Abbildung 5.5:** Unterschiedliches Fehlerverhalten bei der NGP- und der CIC-Interpolation. Dargestellt sind Systeme mit 10.000 bzw. 50.000 Teilchen.

miteinander interagieren oder die auf die Gitterpunkte interpolierten Ladungen. Bei einer solch feinen Untergliederung ist der Zeitaufwand des Verfahrens wesentlich größer als bei der expliziten Methode, so dass das hier verwendete Verfahren keinerlei Vorteile mehr bietet.

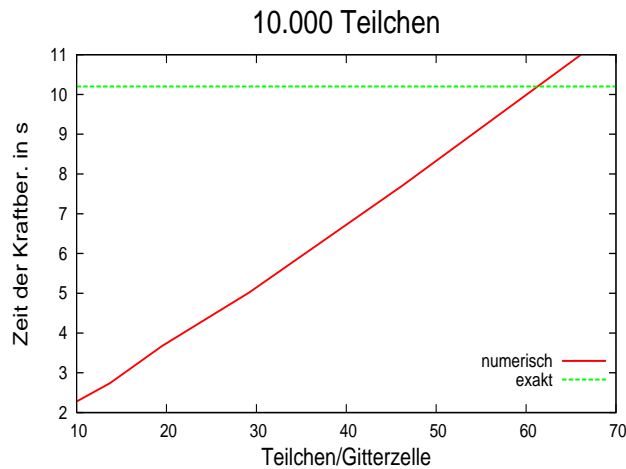


**Abbildung 5.6:** Darstellung des Fehlers in den Kräften, verursacht durch die Diskretisierung, bei einer Dichte von 0.001, der CIC-Interpolation und 125 Nachbarzellen für unterschiedlich viele Teilchen in der Simulation.

Das Ergebnis für die Verwendung der finiten Differenzenschemata lässt sich direkt von den Systemen mit periodischen Randbedingungen auf die offenen Systeme übertragen. Auch hier liefert ein finites Differenzenschema höherer Ordnung bessere Ergebnisse bei nur geringfügiger Steigerung des Aufwands. Siehe hierzu auch Abbildung 5.6.

Auch wenn offene Systeme und Systeme mit periodischen Randbedingungen unterschiedliche Fehlerverhalten aufweisen, sind doch in beiden Fällen die optimalen Parametereinstellungen dieselben: Das CIC-Schema sollte dem NGP-Schema vorgezogen werden und das implementierte finite Differenzenschema höherer Ordnung liefert bessere Ergebnisse als das implementierte Schema niedrigerer Ordnung. Da das Gesamtsystem in kubische Gitterzellen unterteilt wird, ist es bei einer gegebenen Teilchenanzahl unmöglich, jede beliebige Teilchenanzahl pro Gitterzelle zu erhalten. Dieser Wert ist lediglich ein Richtwert, wobei es sich gezeigt hat, dass 30 Teilchen pro Gitterzelle ein guter Kompromiss zwischen schneller Laufzeit einerseits und niedrigem Fehler andererseits ist.

Wird zuletzt noch der Aufwand des expliziten Verfahrens mit dem hier implementierten Verfahren ohne Verwendung des Kompressionsverfahrens verglichen, der Wavelet Transformation also, so ist der Zeitbedarf des hier verwendeten Verfahrens deutlich geringer. Ausnahme bildet lediglich der Fall bei Systemen mit periodischen Randbedingungen, bei dem alle Gitterzellen Nachbarzellen sind und somit das gesamte System als Nahfeld definiert ist. In diesem Fall erzeugt die hier vorgestellte Methode einen Overhead (Abbildung 5.7).



**Abbildung 5.7:** Vergleich der Dauer eines Zeitschrittes des expliziten Particle-Particle Verfahrens und der hier vorgestellten numerischen Methode.

## 5.2 Kompressionsverhalten

Die Wavelet-Transformation kann durch die Art des Mutterwavelets, den Schwellenwert, auch Threshold genannt, die Art des Threshold-Verfahrens (Thresholding) und die Transformationstiefe (Level) gesteuert werden. In dieser Arbeit wurden ausschließlich die von Daubechies entdeckten kompakten, orthogonalen



Wavelets mit verschiedenem Grad der Glättung verwendet. Im weiteren Verlauf werden diese Wavelets mit *Daub2*, *Daub4*, *Daub12* und *Daub20*<sup>2</sup> bezeichnet.

Wurde in der Simulation eine Anzahl von Gitterzellen bestimmt, sind dadurch die Anzahl der Gitterpunkte und die Matrixdimensionen festgelegt. Der Abstand der Gitterpunkte zueinander ist allerdings abhängig von der Kantenlänge einer Gitterzelle bzw. der Kantenlänge der Simulationsbox, welche durch die Teilchenanzahl  $N$  und der Teilchendichte  $\rho$  der MD-Simulation definiert wird. Die Kantenlänge einer Gitterzelle ist jedoch nur ein Skalierungsfaktor, so dass bei der Untersuchung der Kompressionsrate der Wavelet-Transformation immer eine Einheitsgitterzellenlänge von 1 angenommen wurde. Dies bedeutet, dass der Threshold von 0.1 bei einer Gitterzellenlänge von 1 daher dem Schwellenwert von 0.01 bei einer Gitterzellenlänge von 10 entspricht.

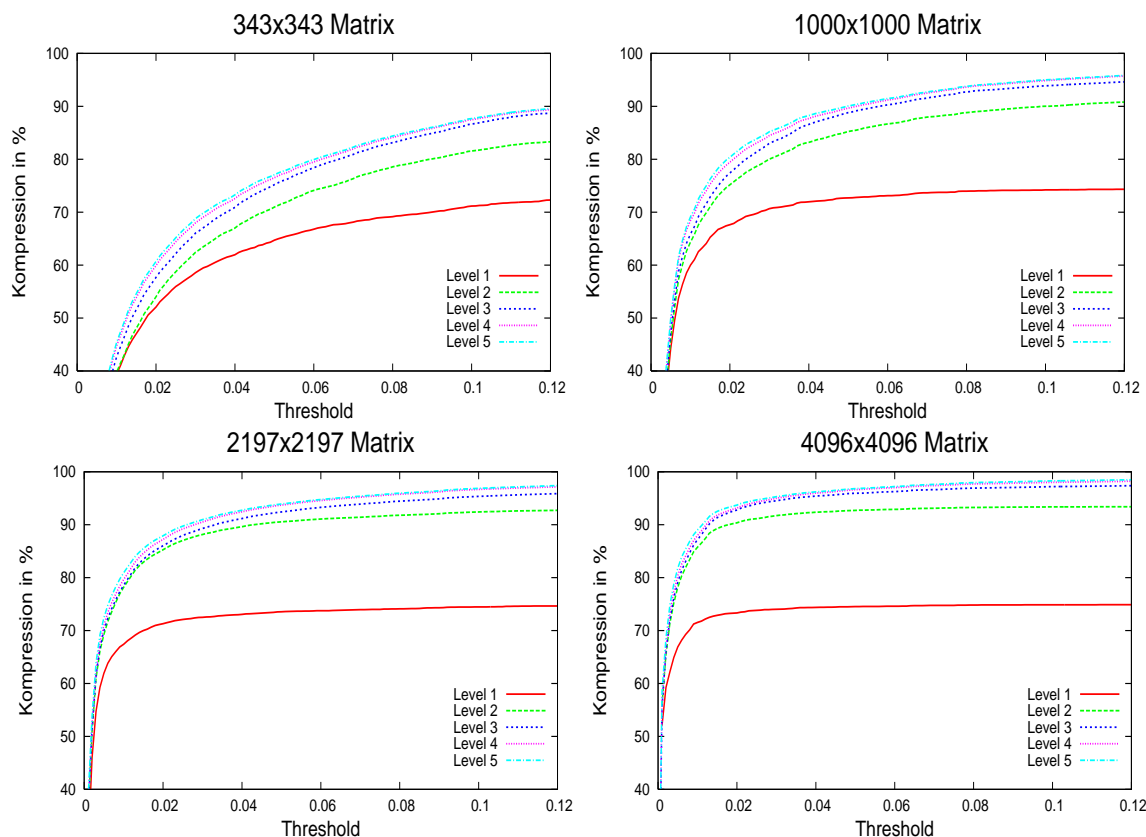
Den größten Einfluss auf das Ergebnis der Wavelet-Transformation hat die Wahl des Thresholds (Schwellenwert). Daher ist es sinnvoll, die Kompressionsrate besonders in Abhängigkeit des Schwellenwertes zu setzen. Die Art des Threshold-Verfahrens beeinflusst nur den eintretenden Fehler. Die Kompressionsrate ist bei beiden Verfahren dieselbe, da bei beiden Verfahrensarten, ausgehend vom Schwellenwert, dieselben Elemente in der Transformationsmatrix zu Null gesetzt werden. Somit hängt die Kompressionsrate nur von der Transformationstiefe (Level), dem Mutterwavelet und vor allem dem Threshold ab.

Unabhängig von der Transformationstiefe, dem benutzten Mutterwavelet und den Dimensionen der Matrix wächst die Kompressionsrate mit zunehmendem Schwellenwert bis zu einer Sättigung an. Da die Einflussgrößen Mutterwavelet und Level nicht voneinander abhängen, ist die Reihenfolge der Bearbeitung der Parameter nicht von Bedeutung.

Zunächst wird versucht, die Transformationstiefe, also den Level, festzulegen. Hierbei ist zu beobachten, dass ein höherer Level eine größere Kompressionsrate zur Folge hat. Da sich der Verlauf der Kompressionsrate mit steigendem Level asymptotisch einer Logarithmus-Funktion annähert und jeder weitere Level lediglich einen Mehraufwand bei der Transformation bedeutet, ist es ratsam, den Level nicht größer als 5 zu wählen. Eine Transformationstiefe von 4 ist völlig ausreichend bei dieser Art von Anwendung. Auffallend ist: Je größer die Matrix ist, desto schneller steigt die Kompressionsrate an. Zu sehen ist dies in Abbildung 5.8. Da die Kompressionsraten bei offenen Systemen nur minimal höher liegen als bei Systemen mit periodischen Randbedingungen, sehe ich davon ab, für beide Fälle die Ergebnisse darzustellen.

---

<sup>2</sup>Im Anhang A sind diese Wavelets mit ihren jeweiligen Skalierungsfunktionen dargestellt.

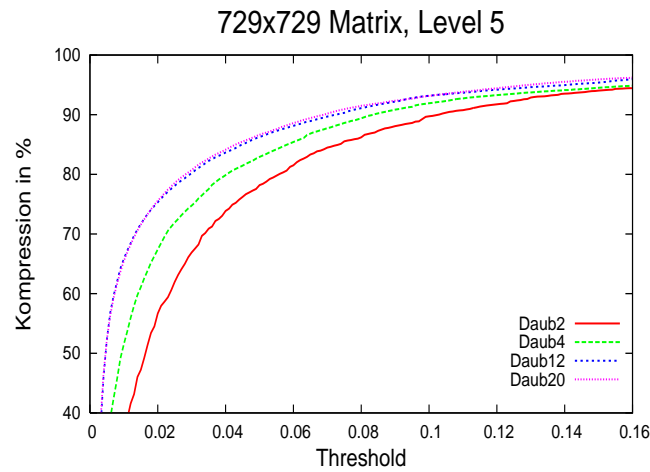


**Abbildung 5.8:** Untersuchung des Einflusses der Transformationstiefe auf die Kompressionsrate bei einigen ausgewählten Matrixgrößen, wobei für alle vier Grafiken *Daub12* als Mutterwavelet verwendet wurde.

Wird das Verhalten der Kompressionsrate bei Transformationen mit verschiedenen Mutterwavelets untersucht, so ist erkennbar, dass mit glatter werdenden Mutterwavelet die Kompressionsrate ansteigt. Da es kaum Unterschiede in der Kompressionsrate bei einer Verwendung von *Daub12* bzw. *Daub20* als Mutterwavelet gibt und ein Mutterwavelet geringerer Ordnung einen geringeren Aufwand verursacht, ist von den hier untersuchten Mutterwavelets *Daub12* für die Anwendung am geeignetsten.

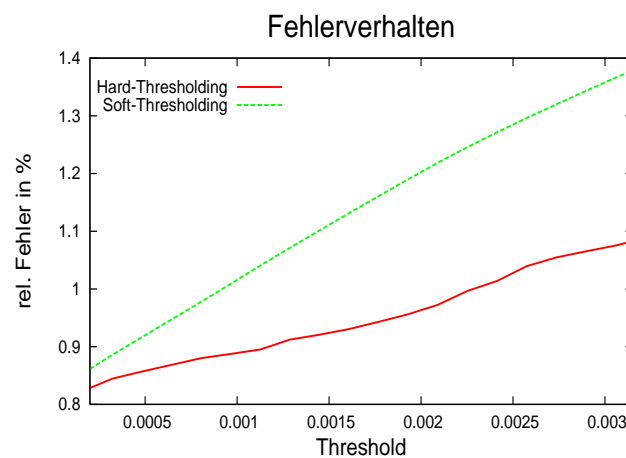
### 5.3 Gesamtfehler

In den vorangegangenen Untersuchungen wurden für fast alle Parameter optimale Einstellungen bestimmt. Gewählt werden müssen lediglich noch die Art des Threshold-Verfahrens (Hard- oder Softthresholding) und der Wert des Thresholds.



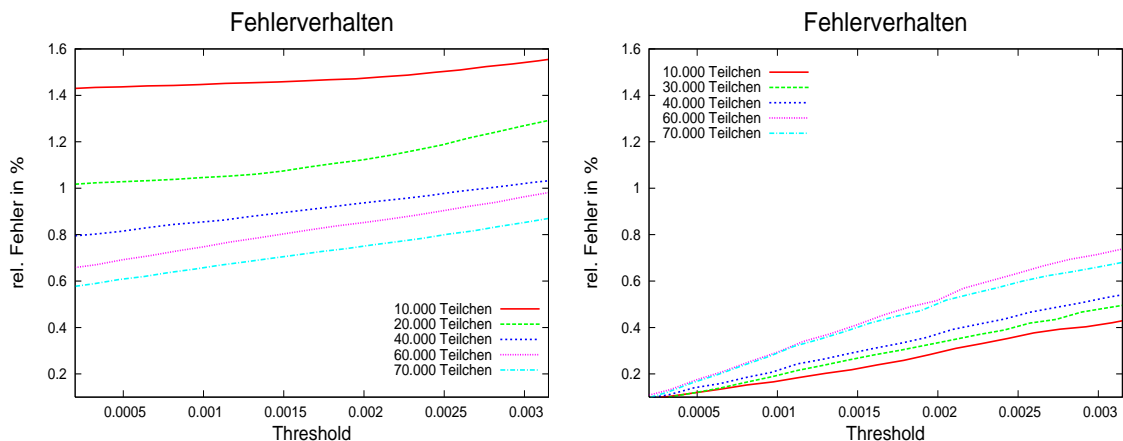
**Abbildung 5.9:** Untersuchung des Einflusses der verschiedenen Wavelettypen auf die Kompressionsrate bei einer Matrixgröße von 729x729, wobei eine Transformationstiefe von 5 verwendet wurde.

Das Soft-Thresholding geht davon aus, dass die Eingangsdaten verrauscht sind, d.h. ein Rauschsignal überlagert das eigentliche Signal. Mit diesem Verfahren soll der Rauschanteil des Signals herausgefiltert werden. In der Praxis bedeutet dies, dass die diskreten Werte des Eingangssignals um die Varianz des Rauschsignals verringert werden. Geht es um das Herausfiltern der Rauschanteile, so mag das Soft-Thresholding gewisse Vorteile gegenüber dem Hard-Thresholding haben. In dem hier vorliegenden Fall geht es allerdings um Matrizenkompression, bei dem das Hard-Thresholding das geeignetere Verfahren ist. In Abbildung 5.10 wird das Soft- und das Hard-Thresholding für ein System mit 30.000 Teilchen gegenübergestellt.



**Abbildung 5.10:** Gegenüberstellung von Soft- und Hard-Thresholding bei einem offenen System mit 30.000 Teilchen.

Das Verhalten des Gesamtfehlers in Abhängigkeit von dem Thresholdwert ist für offene Systeme und für Systeme mit periodischen Randbedingungen sehr ähnlich. In beiden Fällen steigt der Fehler mit höherem Threshold fast linear an. Im Falle der offenen Systeme ist der Fehler geringer als bei Systemen mit periodischen Randbedingungen. Dies ist nicht überraschend, da bei Auslassung der Wavelet Transformation in dem Verfahren bereits ein geringerer Fehler verursacht wird.



**Abbildung 5.11:** Durchschnittlicher Gesamtfehler für Systeme mit periodischen Randbedingungen in der linken und für offene Systeme in der rechten Grafik. Verwendet wurde das Hard-Thresholding.

Der Wert des Thresholds kann nun bestimmt werden, indem eine Fehlerschranke vorgegeben und der größte Thresholdwert ausgewählt wird, bei dem das System einen Fehler unterhalb dieser Schranke aufweist.

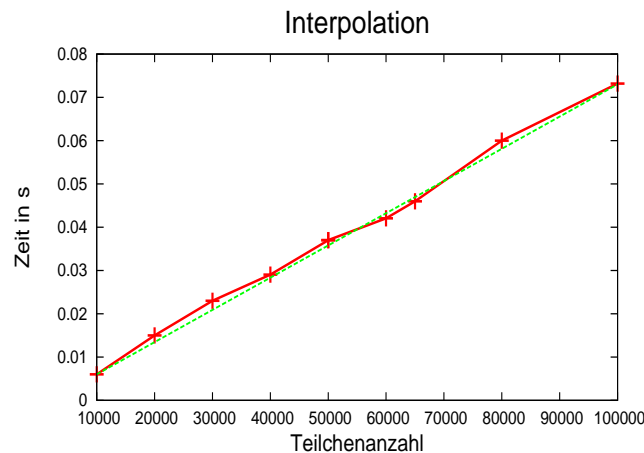
## 5.4 Laufzeitverhalten

Die Kraftberechnung setzt sich aus verschiedenen Unteraufgaben zusammen:

1. Die Interpolation der Ladungen auf die Gitterpunkte
2. Die Matrix-Vektor Multiplikation mit der Wavelet-Transformation (Hin- und Rücktransformation des Ladungs- bzw. Potentialvektors)
3. Die eigentliche Kraftberechnung, bestehend aus der Berechnung des Nahfeldes und der Interpolation des globalen Potentials auf die Teilchen mit der Nahfeldkorrektur

Zu untersuchen ist, welches Verhalten die einzelnen Teilaufgaben aufweisen und damit verbunden, welches Gesamtverhalten der Algorithmus besitzt. Hierzu wird eine feste Fehlerschranke von 1% bei Systemen mit periodischen Randbedingungen bzw. von 0.4% bei offenen Systemen eingeführt.

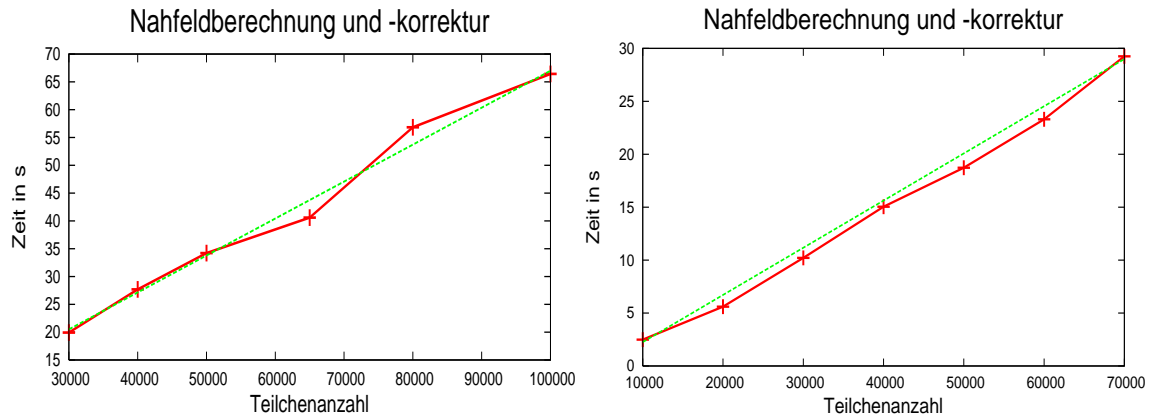
Da die Interpolation der einzelnen Ladungen auf die Gitterpunkte lokal durchgeführt wird, liegt hier ein  $O(N)$  Verhalten vor. Das Laufzeitverhalten ist unabhängig davon, ob ein System mit periodischen Randbedingungen oder ein offenes System vorliegt, da in beiden Fällen die Ladungen aller Teilchen auf dieselbe Weise auf die Gitterpunkte interpoliert werden.



**Abbildung 5.12:** Das Laufzeitverhalten der Interpolation ist nur abhängig von der Teilchenanzahl  $N$ , nicht aber von der Art des Systems.

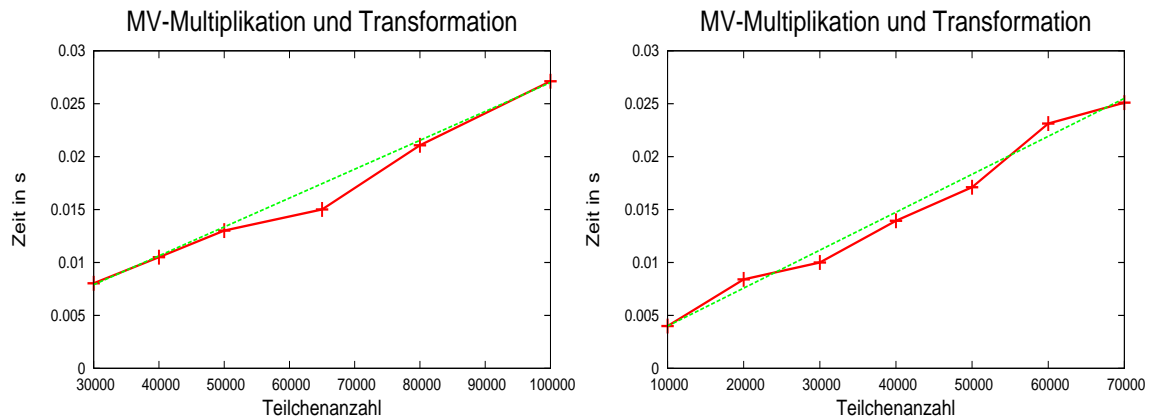
Die Wavelet-Transformation weist ein lineares Verhalten auf, wie bereits in Kapitel 4.4.3 erläutert wurde. Das Nahfeld wird durch die Anzahl von Nachbar-gitterzellen definiert, wobei sich in jeder Gitterzelle durchschnittlich eine konstante Anzahl von Teilchen befindet. Somit ist das Nahfeld unabhängig von  $N$ . Der Zeitbedarf zur Berechnung des Nahfeldes und seine Korrektur ist mit einem linearen Aufwand verbunden. Da die Anzahl der Teilchen aus dem Nahfeld für Randzellen von offenen Systemen wesentlich geringer ist als für die entsprechenden Gitterzellen bei Systemen mit periodischen Randbedingungen, benötigen offene Systeme weniger Zeit für die Berechnung des Nahfeldes. Dargestellt ist dies in Abbildung 5.13.

Diese Ergebnisse überraschen nicht. Interessant ist das Verhalten der Matrix-Vektor Multiplikation. Ohne eine Kompression würde hier ein  $O(N^2)$  Verhalten auftreten, doch mit einer Kompression liegt auch bei dieser Unteraufgabe ein linearer Zeitbedarf vor. Siehe hierzu Abbildung 5.14. An dieser Stelle sei angemerkt, dass die Hintransformation des Ladungsvektors, die Durchführung der Matrix-



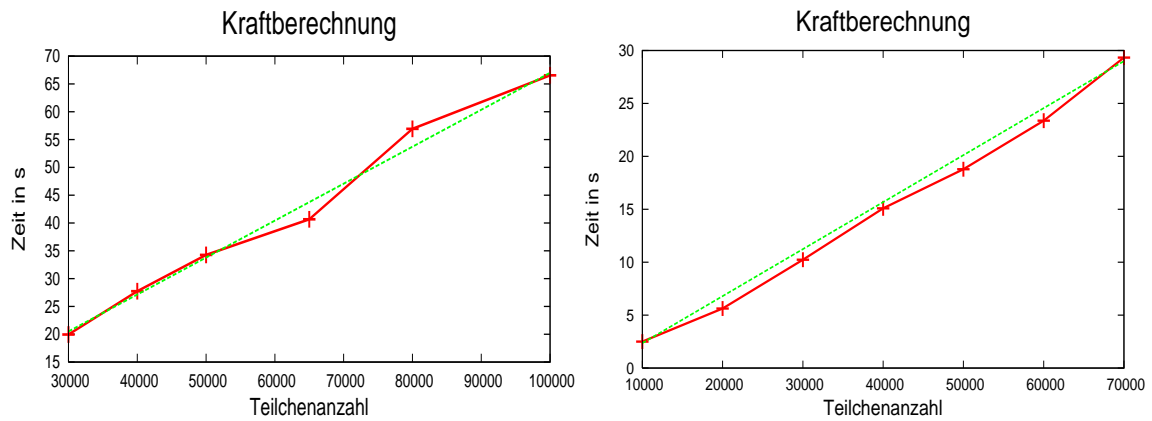
**Abbildung 5.13:** Das Laufzeitverhalten der Berechnung des Kraftanteils aus dem Nah- und dem Fernfeld in Abhängigkeit zur Teilchenanzahl: im linken Bild für Systeme mit periodischen Randbedingungen, im rechten für offene Systeme.

Vektor Multiplikation und die Rücktransformation des Potentialvektors nur einen sehr geringen Bruchteil der Gesamtlaufzeit ausmachen, während der größte Zeitaufwand in der Berechnung des Nahfeldes liegt.



**Abbildung 5.14:** Laufzeitverhalten der Wavelet-Transformation (Hin- und Rücktransformation) und Matrix-Vektor Multiplikation: Die linke Grafik zeigt das Verhalten für Systeme mit periodischen Randbedingungen, die rechte das für offene Systeme.

Da alle Unteraufgaben ein  $O(N)$  Verhalten aufweisen, besitzt ein Zeitschritt, die gesamte Kraftberechnung, ebenfalls einen linearen Aufwand.



**Abbildung 5.15:** Zeitbedarf für einen Zeitschritt von Systemen mit periodischen Randbedingungen (links) bzw. von offenen Systemen (rechts).





# Kapitel 6

## Zusammenfassung und Ausblick

Bei dem hier entwickelten Verfahren handelt es sich um eine Wavelet basierte Summation von langreichweitigen Coulomb-Paarwechselwirkungen. Der Grundgedanke ist, das Simulationsgebiet mit Hilfe eines festen Gitters zu diskretisieren und die Berechnung des Potentials  $\tilde{\Phi}$  auf den Gitterpunkten als eine Matrix-Vektor Multiplikation darzustellen, wobei die Matrix  $\tilde{A}$  aus den inversen Gitterpunktabständen und der Vektor  $\tilde{q}$  aus den zeitabhängigen Ladungen der Gitterpunkte besteht.

$$\tilde{\Phi} = \tilde{A} \cdot \tilde{q}$$

Durch eine Wavelet-Transformation, die mit einem Threshold-Verfahren gekoppelt ist (ausgedrückt durch den Operator  $T$ , der in Kapitel 4.7 bereits definiert wurde), wird die Matrix in eine dünn besetzte Matrix  $\hat{A}$  transformiert. Dies bedeutet für die Simulation, dass in jedem Zeitschritt der Ladungsvektor transformiert, die Matrix-Vektor Multiplikation ausgeführt und das Ergebnis, das globale Potential auf den Gitterpunkten  $\hat{\Phi}$ , wieder zurück transformiert werden muss.

$$\hat{\Phi} = \hat{A}_t \cdot \hat{q} \quad , \quad \hat{A}_t = T(WAW^T)$$

Die Matrix  $W$  bezeichnet hierbei die Transformationsmatrix. Nähere Erläuterungen zu dieser Matrix sind in Kapitel 4.6 zu finden.

Um eine höhere Genauigkeit zu erreichen, wird das Nahfeld durch explizite Paarwechselwirkungen der Teilchen selber berechnet, so dass eine Korrektur des globalen Potentials auf den Gitterpunkten notwendig wird. Die Kräfte, welche aus dem Fernfeld herrühren, werden mit Hilfe des globalen Potentials durch ein finites Differenzenschema berechnet und auf die Teilchen interpoliert.

Dieses Verfahren kann durch einige Parameter gesteuert werden. Eine Auflistung der Parameter ist in Tabelle 6.1 gegeben. Werden die Parameter geeignet vorgegeben, so besitzt das Verfahren bei einer vorgegebenen Fehlerschranke einen Aufwand der Ordnung  $O(N)$ .

Parameter	Einstellung
Anzahl der Gitterzellen	$\sqrt[3]{N}/30$
Anzahl der Nachbarzellen	125
Interpolationsart	CIC
Differenzen-Schema	4. Ordnung
Basiswavelet	<i>Daub12</i>
Transformationstiefe	Level 4
Threshold Verfahren	Hard-Thresholding
Threshold (Wert): period. S. : offene S.	$2.57 \cdot 10^{-8}N + 9.28 \cdot 10^{-4}$ $-2.33 \cdot 10^{-8}N + 3.03 \cdot 10^{-3}$

**Tabelle 6.1:** Auflistung der Verfahrensparameter mit ihren Voreinstellungen, wobei  $N$  die Anzahl der Teilchen in der Simulation ist. Untersucht worden sind die einzelnen Parametereinstellungen für Systeme von 10.000 bis 100.000 Teilchen.

Das hier entwickelte Verfahren kann noch in vielerlei Hinsicht erweitert werden. Denkbar sind Alternativen zu bestehenden Teilaufgaben wie z.B. andere Interpolations-Schemata, andere Mutterwavelets oder andere finite Differenzen-Schematas. Eine weitere Möglichkeit der Erweiterung besteht darin, neben offenen Systemen und Systemen mit periodischen Randbedingungen auch Mischformen zuzulassen.

Da zur Zeit die Parameter per Default so eingestellt sind, dass eine Fehler-schranke von 1% bei Systemen mit periodischen Randbedingungen bzw. von 0,4% bei offenen Systemen im Mittel nicht überschritten wird, besteht für die Zukunft die Aufgabe, einen Algorithmus zu entwickeln, der die optimalen Werte für die Parameter ermittelt, abhängig von der Art des vorliegenden Problems und einer von dem Benutzer vorgegebenen Fehler- und/oder Zeitschranke.

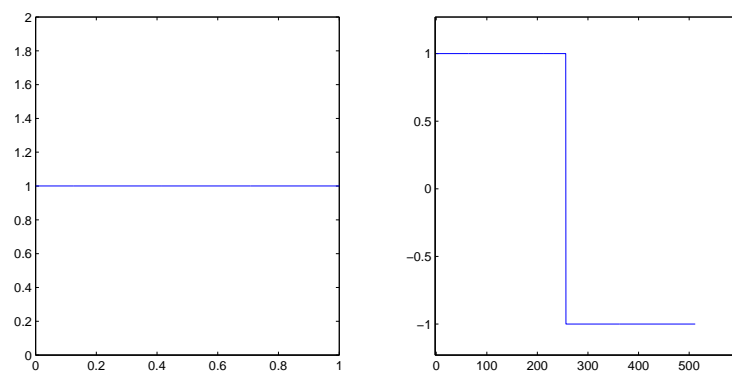
Zur Zeit wird daran gearbeitet, dieses Verfahren zusammen mit anderen Verfahren zur Kraftberechnung in MD-Simulationen in eine Programm-Bibliothek einzubinden.

# ANHANG

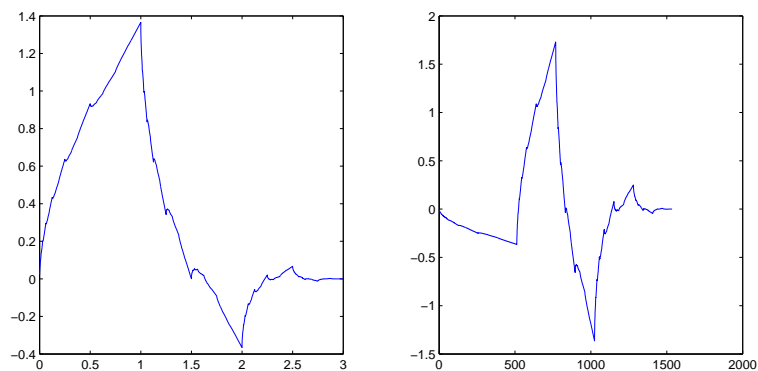


# Anhang A

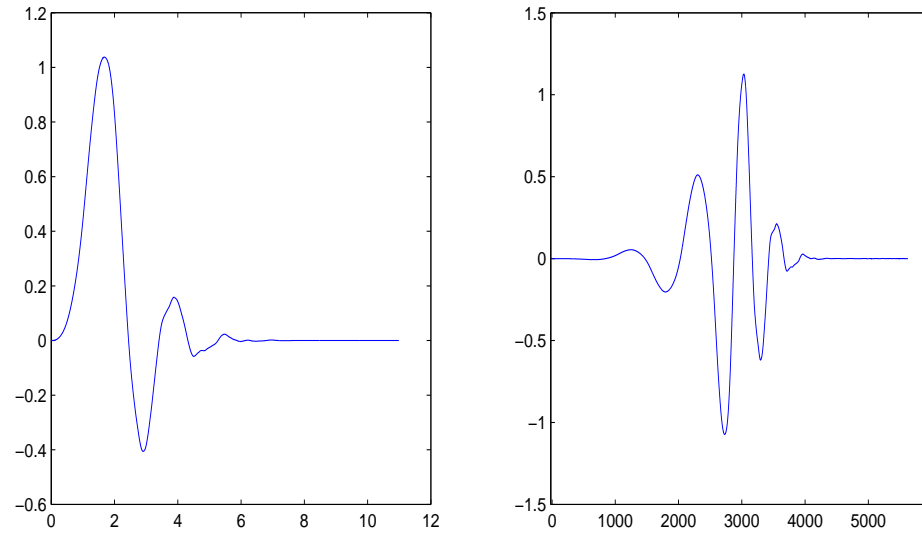
## Verwendete Wavelets



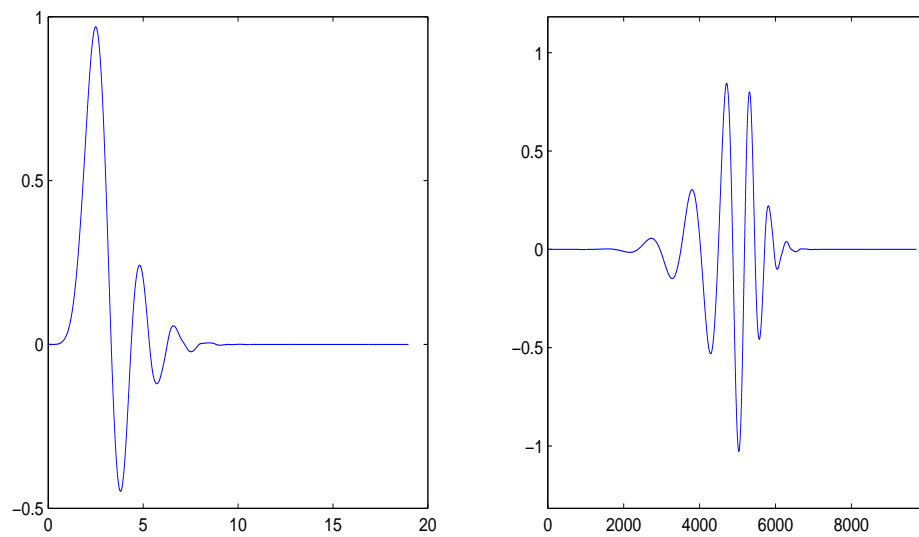
**Abbildung A.1:** Daubechies 2-Wavelet: Skalierungsfunktion (links) und Mutterwavelet (rechts)



**Abbildung A.2:** Daubechies 4-Wavelet: Skalierungsfunktion (links) und Mutterwavelet (rechts)



**Abbildung A.3:** Daubechies 12-Wavelet: Skalierungsfunktion (links) und Mutterwavelet (rechts)



**Abbildung A.4:** Daubechies 20-Wavelet: Skalierungsfunktion (links) und Mutterwavelet (rechts)

	Waveletkoeffizienten $h_k$ der Daubechies-Wavelets			
Ordnung $p$	1	2	6	10
$k = 0$	$1/\sqrt{2}$	$(1 + \sqrt{3}/(4\sqrt{2}))$	0.111540743350	0.026670057901
1	$1/\sqrt{2}$	$(3 + \sqrt{3}/(4\sqrt{2}))$	0.494623890398	0.188176800078
2		$(3 - \sqrt{3}/(4\sqrt{2}))$	0.751133908021	0.527201188932
3		$(1 - \sqrt{3}/(4\sqrt{2}))$	0.315250351709	0.688459039454
4			-0.226264693965	0.281172343661
5			-0.129766867567	-0.249846424327
6			0.097501605587	0.195946274377
7			0.027522865530	0.127369340336
8			-0.031582039318	0.093057364604
9			0.000553842201	-0.071394147166
10			0.004777257511	-0.029457536822
11			-0.001077301085	0.033212674059
12				0.003606553567
13				-0.010733175483
14				0.001395351747
15				0.001992405295
16				-0.000685856695
17				-0.000116466855
18				0.000093588670
19				-0.000013264203

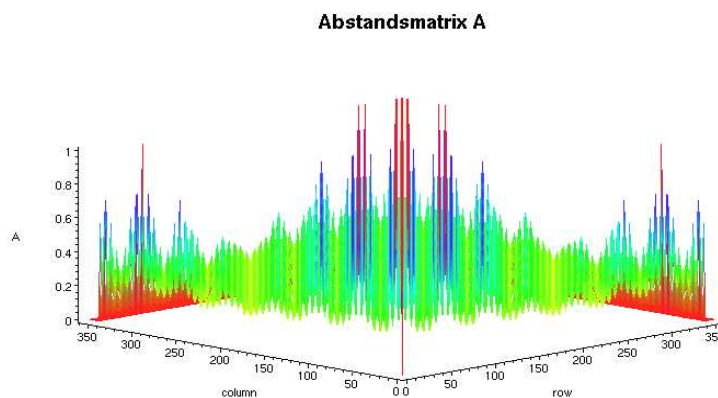
**Tabelle A.1:** Auflistung der Waveletkoeffizienten  $h_k$  der Daubechies Wavelets der hier verwendeten Ordnungen 1, 2, 6, und 10.



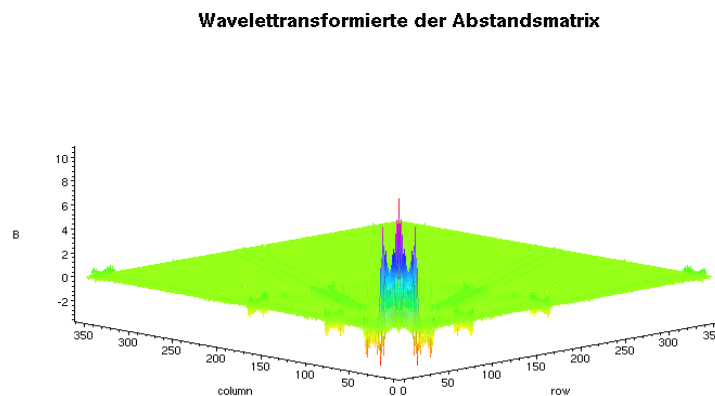


# Anhang B

## Abstandsmatrix

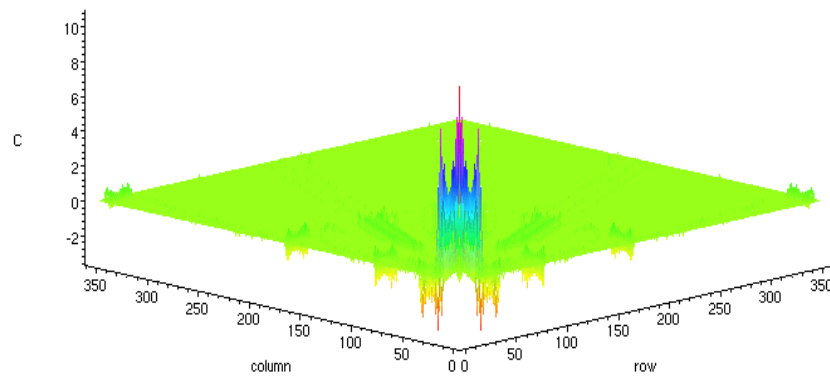


**Abbildung B.1:** Abstandsmatrix für Systeme mit periodischen Randbedingungen: Die letzten Zeilen und Spalten sind mit Nullen aufgefüllt, um die Wavelet-Transformation zu ermöglichen.



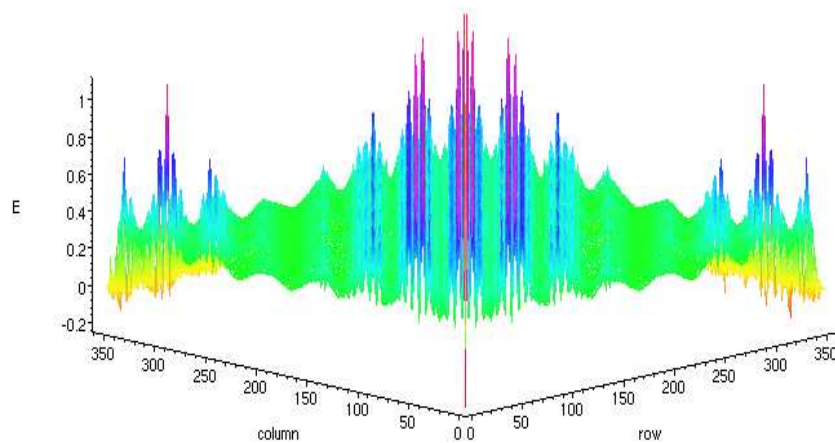
**Abbildung B.2:** Wavelettransformierte der Abstandsmatrix: Die meisten Koeffizienten sind nahe Null, in nur wenigen Koeffizienten sammeln sich die Beiträge.

### Wavelettransformierte nach Hard-Thresholding



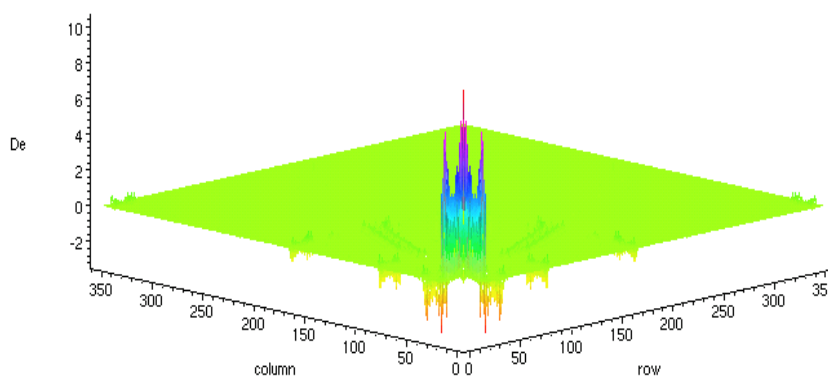
**Abbildung B.3:** Wavelettransformierte Abstandsmatrix nach Anwendung des Hard-Thresholding: Koeffizienten kleiner dem Threshold (hier 0.2) wurden zu Null gesetzt.

### Rücktransformierte nach Hard-Thresholding



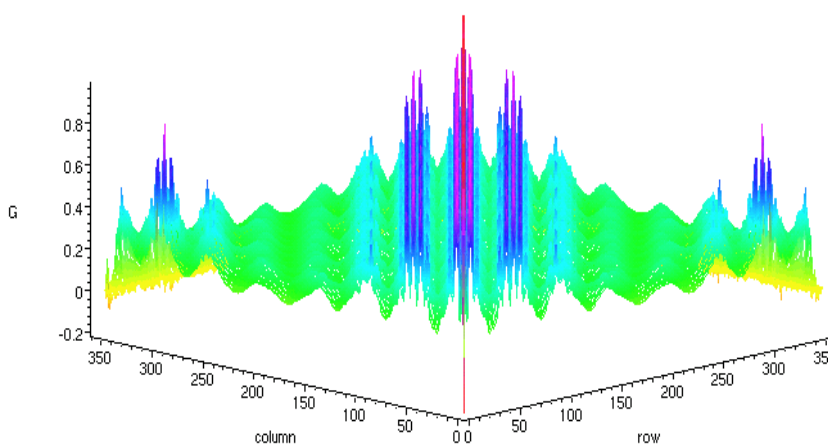
**Abbildung B.4:** Rücktransformierte Abstandsmatrix nach dem Hard-Thresholding : Der wellenförmige Charakter der Abstandsmatrix ist noch vorhanden, allerdings sind die einzelnen Wellen nicht mehr so scharf voneinander getrennt.

### Wavelettransformierte nach Soft-Thresholding

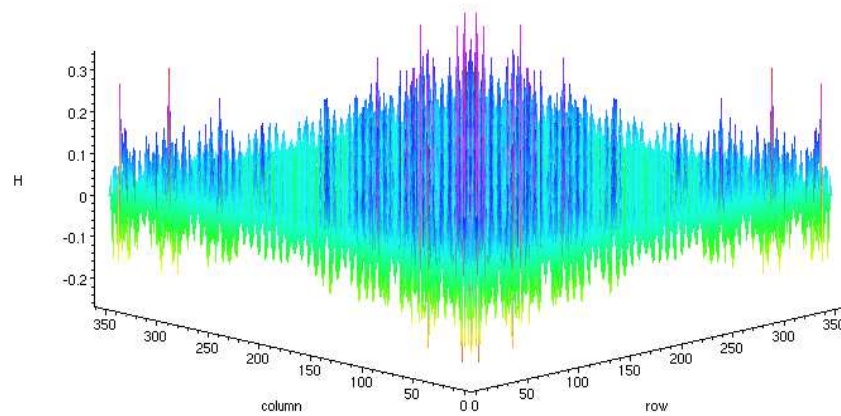


**Abbildung B.5:** Wavelettransformierte Abstandsmatrix nach Anwendung des Soft-Thresholding: Die Koeffizienten kleiner dem Threshold (hier 0.2) wurden zu Null gesetzt, alle anderen Koeffizienten wurden um den Thresholdwert verkleinert bzw. im Negativen vergrößert.

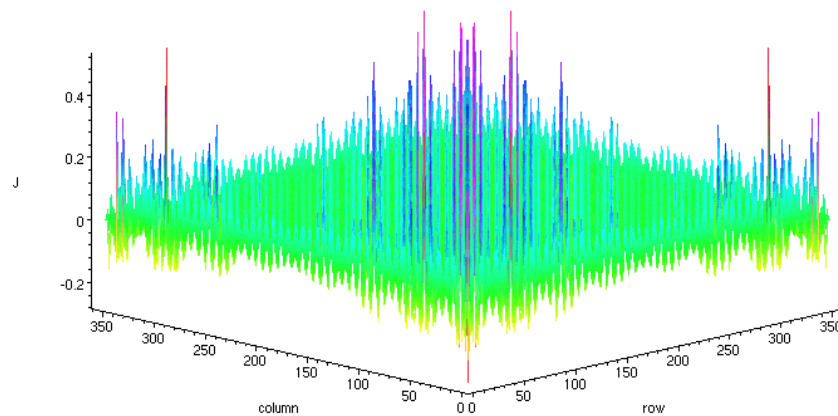
### Rücktransformierte nach Soft-Thresholding



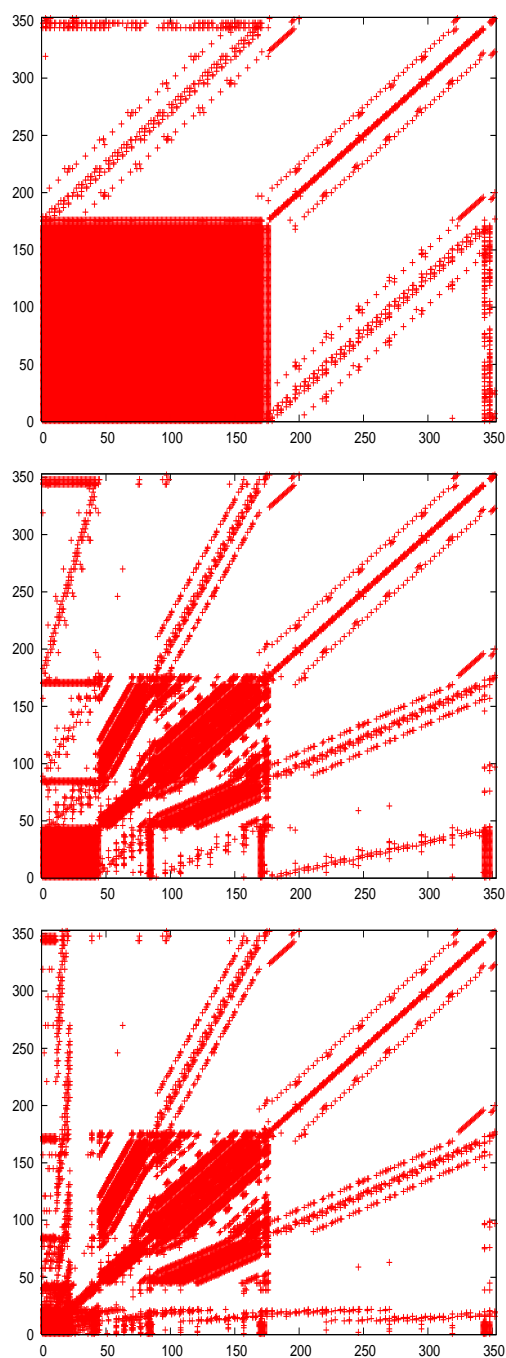
**Abbildung B.6:** Rücktransformierte Abstandsmatrix nach dem Soft-Thresholding: Der wellenförmige Charakter der Abstandsmatrix ist noch vorhanden, die einzelnen Wellen sind jedoch etwas abgeklungen.

**Fehler Hard-Thresholding**

**Abbildung B.7:** Verteilung des durch das Hard-Thresholding verursachten Fehlers: Das Threshold-Verfahren wirkt sich am stärksten auf die Koeffizienten der Abstandsmatrix aus, welche die Abstände der Teilchen aus dem Nahfeld repräsentieren.

**Fehler Soft-Thresholding**

**Abbildung B.8:** Verteilung des durch das Soft-Thresholding verursachten Fehlers: Das Threshold-Verfahren wirkt sich am stärksten auf die Koeffizienten der Abstandsmatrix aus, welche die Abstände der Teilchen aus dem Nahfeld repräsentieren.



**Abbildung B.9:** Die Elemente der wavelettransformierten Abstandsmatrix ungleich Null bei einer Transformationstiefe von 1, 3 und 5: Durch die Wavelet-Transformation und das Threshold-Verfahren wird die vollbesetzte Abstandsmatrix in eine dünn besetzte Matrix konvertiert. Die Matrix hat eine Streifen-Struktur, die mit jedem Level verstärkt wird.



# Anhang C

## Quellcode - FWT

```
!*****!
!*****          SUBROUTINE wt1          *****!
!*****!
!***** Subroutine is the controllstructur of the FWT *****!
!*****!
!*** Parameter:  - a          Datavector          ***!
!***              - n          Length              ***!
!***              - level     Depth of transformation ***!
!***              - isign     Flag for (back-)transformation ***!
!*****!
subroutine wt1(a,n,level,isign)

    integer, intent(in) :: n, level, isign
    real(kind=rp),dimension(n), intent(inout) :: a
    integer :: nn, i

    if(n<4) return
    if(isign>=0) then
        nn=n
        do i=1,level
            call pwt(a,nn,isign)
            nn=nn/2
        end do
    else
        nn=n/(2**(level-1))
        do i=1,level
            call pwt(a,nn,isign)
            nn=nn*2
        end do
    end if
end subroutine wt1
```

```

!*****!
!*****      SUBROUTINE pwt      *****!
!*****!
!*****      Subroutine transforms a vector (FWT)      *****!
!*****!
subroutine pwt(a,n,isign)

  integer, intent(in) :: n, isign
  real(kind=rp),dimension(n), intent(inout) :: a
  real(kind=rp),dimension(n) :: wksp
  integer :: i, j, ii, nh, nmod

  nh=n/2
  wksp=0.0
  if(isign>=0) then !*****      Transformation      *****!
    ii=1
    do i=1,n,2
      do j=1,ncof
        nmod=modulo(i+j-1,n+1)+(i+j)/(n+2)
        wksp(ii)=wksp(ii)+cc(j)*a(nmod)
        wksp(ii+nh)=wksp(ii+nh)+cr(j)*a(nmod)
      end do
      ii=ii+1
    end do
  else !*****      Backtransformation      *****!
    ii=1
    do i=1,n,2
      do j=1,ncof
        nmod=modulo(i+j-1,n+1)+(i+j)/(n+2)
        wksp(nmod)=wksp(nmod)+cc(j)*a(ii)+cr(j)*a(ii+nh)
      end do
      ii=ii+1
    end do
  end if

  do i=1,n
    a(i)=wksp(i) !*****      Save      *****!
  end do
end subroutine pwt

```

Die Vektoren *cc* bzw. *cr* beinhalten die Koeffizienten des Wavelets  $h_k$  (vgl. Tabelle A.1) bzw. die Koeffizienten der Skalierungsfunktion  $g_k$ .



# Anhang D

## Notationen

$\mathbb{N}, \mathbb{N}_0$	Menge der natürlichen Zahlen, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$
$\mathbb{Z}$	Menge der ganzen Zahlen
$\mathbb{R}$	Körper der reellen Zahlen
$\mathbb{C}$	Körper der komplexen Zahlen
$L^p(\mathbb{R})$	$L^p(\mathbb{R}) \stackrel{\text{def}}{=} \left\{ f : \mathbb{R} \rightarrow \mathbb{R} \mid \int_{-\infty}^{\infty}  f(x) ^p dx < \infty \right\}, 1 \leq p \leq \infty$
$\ell^p(\mathbb{R})$	$\ell^p(\mathbb{R}) \stackrel{\text{def}}{=} \left\{ \{a_k\}_{k \in \mathbb{Z}} \mid \sum_{k \in \mathbb{Z}}  a_k ^p < \infty \right\}, 1 < p < \infty$
$\overline{\bigcup_{m \in \mathbb{Z}} V_m}$	$L^2$ -Abschluss der Vereinigung der Menge $V_m$
$\bigcap_{m \in \mathbb{Z}} V_m$	Schnitt der Mengen $V_m$
$\langle \cdot, \cdot \rangle_{L^2}$	Skalarprodukt von $L^2(\mathbb{R})$
$\ \cdot\ _{L^2}$	Norm von $L^2(\mathbb{R})$
$\oplus$	direkte Summe
$P_m f$	Approximation von $f$ auf Skala $m$
$Q_m f$	Detail von $f$ auf Skala $m$
$\varphi$	Skalierungsfunktion
$\psi$	Wavelet
$g_k$	Koeffizienten des Wavelets
$h_k$	Koeffizienten der Skalierungsfunktion
$c_k$	Approximations-Koeffizienten von $f$ (von Kombination mit $\varphi$ )
$d_k$	Detail-Koeffizienten von $f$ (von Kombination mit $\psi$ )
$L_\psi f$	Wavelet-Transformation von Funktion $f$ zum Wavelet $\psi$

$G$	$G : \ell^2(\mathbb{Z}) \rightarrow \ell^2(\mathbb{Z})$ mit $c \mapsto Gc = \left\{ (Gc)_k = \sum_{l \in \mathbb{Z}} g_{l-2k} c_l \right\}$
$G^*$	adjungierter Operator zu $G$ bezüglich des $L^2$ -Skalarproduktes
$H$	$H : \ell^2(\mathbb{Z}) \rightarrow \ell^2(\mathbb{Z})$ mit $c \mapsto Hc = \left\{ (Hc)_k = \sum_{l \in \mathbb{Z}} h_{l-2k} c_l \right\}$
$H^*$	adjungierter Operator zu $H$ bezüglich des $L^2$ -Skalarproduktes

# Literaturverzeichnis

- [1] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, L. G. Pedersen: A smooth particle mesh Ewald method, American Institute of Physics, 1995
- [2] R. W. Hockney und J. W. Eastwood: Computer simulation using particles, Vol. 1, Mc Graw-Hill, New York, 1999
- [3] G. Joos: Lehrbuch der theoretischen Physik, Akademische Verlagsgesellschaft Geest & Portig K.-G., Leipzig, 1964
- [4] G. Sutmann, B. Steffen: A Particle Particle-Multigrid Method for long-range Interaction in Molecular Simulations, Computer Physics Communications, 2005 (in press)
- [5] G. Sutmann: Computersimulationsmethoden in den Naturwissenschaften, Lecture Notes, 2003
- [6] G. Beylkin, R. Coifman, V. Rokhlin: Fast Wavelet Transforms and Numerical Algorithms I, Comm. on Pure and Appl. Math., Vol. 44, 1991
- [7] G. Beylkin: Wavelets and Fast Numerical Algorithms, Lecture Notes, 1993
- [8] G. Beylkin: Wavelets, Multiresolution Analysis and Fast Numerical Algorithms, A draft of INRIA Lecture Notes, 1991
- [9] C. Blatter: Wavelets - Eine Einführung, Vieweg Verlag, Braunschweig, 1998
- [10] W. Bäni: Wavelets - Eine Einführung für Ingenieure, Oldenburg Verlag, München, Wien, 2002
- [11] I. Daubechies: Ten Lectures on Wavelets, SIAM, Philadelphia, 1992
- [12] B. P. Flannery, W. H. Press, S. A. Teukolsky, W. T. Vetterling: Numerical Recipes in Fortran 77: The Art of scientific Computing, Cambridge University Press, 1999
- [13] A. Graps: An Introduction to Wavelets, IEEE Computational Science and Engineering, Los Alamitos, Kalifornien, 1995

- [14] O. Hansen: Einführung in die Theorie und Anwendung der Wavelets, Logos Verlag, Berlin, 2000
- [15] B. B. Hubbard: The world according to Wavelets, A. K. Peters, Ltd, Wellesley, Massachusetts, 1998
- [16] B. B. Hubbard: Wavelets - Mathematik der kleinen Wellen, Birkhäuser Verlag, Berlin, 1997
- [17] G. Kaiser: A friendly Guide to Wavelets, Birkhäuser, Boston, 1994
- [18] F. Keinert: Biorthogonal Wavelets for Fast Matrix Computations, Applied and Computational Harmonic Analysis, 1992
- [19] A. K. Louis, P. Maaß, A. Rieder: Wavelets: Theorie und Anwendungen, Teubner, Stuttgart, 1994
- [20] P. A. Morettin: Wavelets in Statistics, Reviews of The Institute of Mathematics and Statistics of The University of Sao Paulo, Vol. 3, 1997
- [21] Yves Nievergelt: Wavelets made easy, Birkhäuser Verlag, Boston, 1999
- [22] R. Schneider: Multiskalen- und Wavelet-Matrixkompression, B.G. Teubner, Stuttgart, 1998
- [23] J. W. Harris und Horst Stocker: Handbook of Mathematical and Computational Sciences, Springer, New York 1998